Université Saâd DAHLAB de Blida

Faculté des sciences

Département d'informatique

Mémoire Présenté par

BECILA Amira

En vue d'obtenir le diplôme de Master

Domaine : Mathématique et Informatique MI.

Filière : Informatique.

Spécialité : Ingénierie des logiciels.

Sujet :

Mise en œuvre d'un Processus ETL dans un environnement Hadoop Hive

Promoteur : Mr. M.BALA

Soutenue le : devant le jury composé de :

M.	Président
M.	Examinateur
M.	Examinateur

Remerciements

Je remercie DIEU pour les heures de courage, de patience, et de sagesse qu'il m'a inspiré.

Je remercie aussi tous mes enseignants qui ont contribué à ma formation durant mes cinq années d'études, et qui m'ont fait part de leur savoir.

Je remercie mon promoteur Mr.BALA pour avoir dirigé ce travail avec grande attention. Je lui rends hommage pour ses qualités humaines, son intégrité scientifique et sa disponibilité à tout moment. Ses critiques et ses judicieux conseils m'ont été très précieux pour la réalisation de ce mémoire. Merci encore.

Je remercie le jury de m'avoir consacré son temps pour lire mon mémoire.

Enfin, je remercie mes proches et mes amis de m'avoir soutenu tout au long de ce travail.

MERCI A TOUS

Sommaire

Liste des tableaux

Liste des figures

Liste des diagrammes

Liste des acronymes

Selon leur ordre d'apparition

ETL	Extract-Transform-Load
DW	Data Warehouse
ODS	Operational Data Store
OLAP	Online Analytical Processing
SQL	Structured Query Language
SGBD	Système de Gestion de Base de Données
OLTP	Online Transaction Processing
ETLMR	Extract Transform Load MapReduce
HDFS	Hadoop Distributed File System
OS	Operating system
UDF	User Defined Function
ASF	Apache Software Foundation

ملخص

تقوم هذه المذكرة على تنفيذ عملية ال ETL في بيئة Hadoop Hive. هذه البيئة مصممة للتطبيقات الموزعة وإدارة المعطيات المكثفة. تسمح هذه البيئة للتطبيقات بالعمل مع الآلاف من العقد والبيتابايت من المعطيات.

لتأدية وتسريع مخلف مهام عملية ال ETL أي استخراج وتحويل وتحميل المعطيات في مستودع المعطيات، نطبق عليه خوارزمية MapReduce لأنها أكثر تلاؤما مع الحسابات المتوازية والموزعة للمعطيات الكثيفة الحجم. تتمتع بيئتنا بإدماج كامل لهذه الخوارزمية.

الكلمات المفتاحية : مستودع المعطيات، عملية ال ETL ، MapReduce ، Cloud Computing ، Hadoop، Hive، Sqoop، HDFS.

Résumé

Ce mémoire consiste en la mise en œuvre du processus ETL dans un environnement Hadoop Hive. Cet environnement est destiné aux applications distribuées et à la gestion intensive des données. Il permet aux applications de travailler avec des milliers de nœuds et des pétaoctets de données.

Pour performer et accélérer les différentes tâches du processus ETL à savoir l'extraction, la transformation et le chargement des données dans un entrepôt, nous lui appliquons l'algorithme MapReduce qui est le plus adapté pour le calcul parallèle et distribué de données potentiellement très volumineuses. Notre environnement dispose d'une implémentation complète du paradigme MapReduce.

Mots clés : Entrepôt de données, processus ETL, MapReduce, Cloud Computing, Hadoop, Hive, Sqoop, HDFS.

Summary

This thesis consists of the implementation of the ETL process in Hadoop Hive. This environment is designed for distributed applications and management of data intensive. It allows applications to work with thousands of nodes and petabytes of data.

To perform and accelerate the various tasks of ETL process namely the extraction, transformation and loading data into a Data Warehouse, we apply the algorithm MapReduce which is the most suited for parallel and distributed data potentially very large. Our environment has a complete implementation of this algorithm.

Keywords: Data Warehouse, ETL process, MapReduce, Cloud Computing, Hadoop, Hive, Sqoop, HDFS.

Introduction générale

Introduction générale

C'est dans un environnement fortement complexe et hautement concurrentiel qu'évolue la majeure partie des entreprises. Ce climat exige de ces entreprises une surveillance très étroite du marché afin de ne pas se laisser distancer, et cela en répondant le plus rapidement possible aux attentes de leur clientèle et de leurs partenaires.

Les dirigeants doivent prendre les décisions les plus opportunes qui influeront grandement sur la stratégie de l'entreprise et donc sur son avenir.

C'est dans ce contexte que les « **systèmes décisionnels** » ont vu le jour. Ils offrent aux décideurs des informations de qualité sur lesquelles ils pourront s'appuyer pour leur choix décisionnels.

Le système décisionnel dispose du processus d'Extraction, de Transformation et de Chargement, en anglais Extract-Transform-Load (**ETL**) qui comporte trois phases :

1- Extraction des données : c'est la capture des données ayant servi à la gestion du quotidien de l'entreprise n'ayant pas donc des qualités d'analyse.
2- Transformation des données : pour leur donner de la valeur, ces données doivent être nettoyées, homogénéisées, filtrées et agrégées.
3- Chargement des données : consiste à charger les données transformées dans une base appelée « **Entrepôt de données** ».

L'entrepôt de données (**Datawarehouse**), appelé aussi « base de données multidimensionnelle », contient des données prêtes pour l'analyse puisque structurées sous forme d'indicateurs (mesures) et axes d'analyse (dimensions).

Une diversité d'outils d'analyse et de reporting existe aujourd'hui pour exploiter l'entrepôt de données en restituant des rapports et présentant des interfaces d'analyse multidimensionnelles et de datamining.

Contexte

Nous nous intéressons dans ce projet au processus ETL vu sa complexité et son importance dans la réussite d'un projet décisionnel. Parmi tous les aspects ETL, nous nous sommes focalisés dans ce travail sur l'aspect performance en termes d'exécution et temps de réponse.

Problématique

ETL est le moyen qui permet l'alimentation du datawarehouse. Comme les sources de données évoluent sans cesse suite aux événements transactionnels, le processus est chargé de rafraichir l'entrepôt de données. Ce qui fait de lui un processus continu.

Les technologies traditionnelles de ETL rencontrent de nouveaux défis du à l'accroissement des informations. De nos jours, l'entreprise recueille des centaines de gigaoctets de données pour le traitement et l'analyse. Cette grande quantité de données rend ETL extrêmement lent en termes de temps, et pour l'évolution des environnements d'affaires, les utilisateurs ont une demande croissante d'obtenir des données dès que possible.

Pour palier à ce problème, l'utilisation de la parallélisation des tâches est la clé pour améliorer les performances et l'évolutivité de ces défis.

Objectif

Au cours de ses dernières années, de nouvelles technologies de parallélisation de tâches ont vu le jour, parmi elles, le paradigme **MapReduce**. Ce dernier, est un framework de développement informatique, introduit par Google, dans lequel sont effectués des calculs parallèles, et souvent distribués des données potentiellement très volumineuses.

Notre objectif est de performer et d'accélérer les différentes tâches du processus ETL. Pour cela, nous nous intéressons à sa mise en œuvre sous l'environnement **Hadoop Hive** qui intègre le concept MapReduce.

Organisation du mémoire

Afin de répondre à la problématique présentée précédemment, ce mémoire est organisé comme suit :

- Chapitre I : Les systèmes décisionnels
  Dans ce chapitre nous avons défini le système décisionnel et ses principales phases.
- Chapitre II : Le processus ETL et le paradigme MapReduce
  Ce chapitre défini le processus ETL et ses trois étapes (extraction, transformation et chargement des données) ainsi que son importance et sa complexité. Vient ensuite le paradigme MapReduce qui est une solution pour accélérer le travail de ETL.
- Chapitre III : L'environnement Hadoop Hive
  Dans ce chapitre nous avons défini l'environnement Hadoop, ses sous projets MapReduce, Hive et HDFS.
- Chapitre IV : La mise en œuvre du processus ETL sous l'environnement Hadoop Hive
  Ce chapitre englobe les trois premiers chapitres car il consiste en la mise en œuvre du processus ETL sous l'environnement Hadoop Hive.

Chapitre I

Les systèmes décisionnels

1- Introduction

Les applications classiques d'une organisation permettent de **stocker**, **restituer** et/ou **modifier** les données des différents services opérationnels de l'entreprise. Ces différents services possèdent chacun une ou plusieurs applications propres, et les données y sont rarement structurées ou codifiées de la même manière que dans les autres services. Chaque service dispose le plus souvent de ses propres tableaux de bord et il est rare que les indicateurs (par exemple : le chiffre d'affaires sur un segment de clientèle donné) soient mesurés partout de la même manière, selon les mêmes règles et sur le même périmètre même si il est possible d'évaluer l'entreprise.

Pour pouvoir obtenir une vision synthétique de chaque service ou de l'ensemble de l'entreprise, il convient donc que ces données soient **filtrées, croisées** et **reclassées** dans un entrepôt de données central. Cet entrepôt de données va permettre aux responsables de l'entreprise et aux analystes de prendre connaissance des données à un niveau global et ainsi prendre des décisions plus pertinentes, d'où le nom d'**informatique décisionnelle.**

2- Informatique décisionnelle

On qualifie d'informatique décisionnelle (en anglais « **Business intelligence** », parfois appelé tout simplement « **le décisionnel** ») « l'exploitation des données de l'entreprise dans le but de faciliter la prise de décision par les décideurs, c'est-à-dire la compréhension du fonctionnement actuel et l'anticipation des actions pour un pilotage éclairé de l'entreprise. » **[Reinschmidt,Francoise,2000]**. Cette dernière est passée du système d'information de production (gestion) au système d'information décisionnel (pilotage).

« Un système décisionnel (Decision Support System) est l'ensemble des moyens (matériels, logiciels et humains) et techniques permettant de capitaliser le parcours d'une organisation dans un but d'évaluation, d'analyse

et d'aide à la décision. » **[Ponniah, 2001]**. Ce système comporte quatre phases et est conçu selon une architecture à trois niveaux comme le montre la figure suivante :

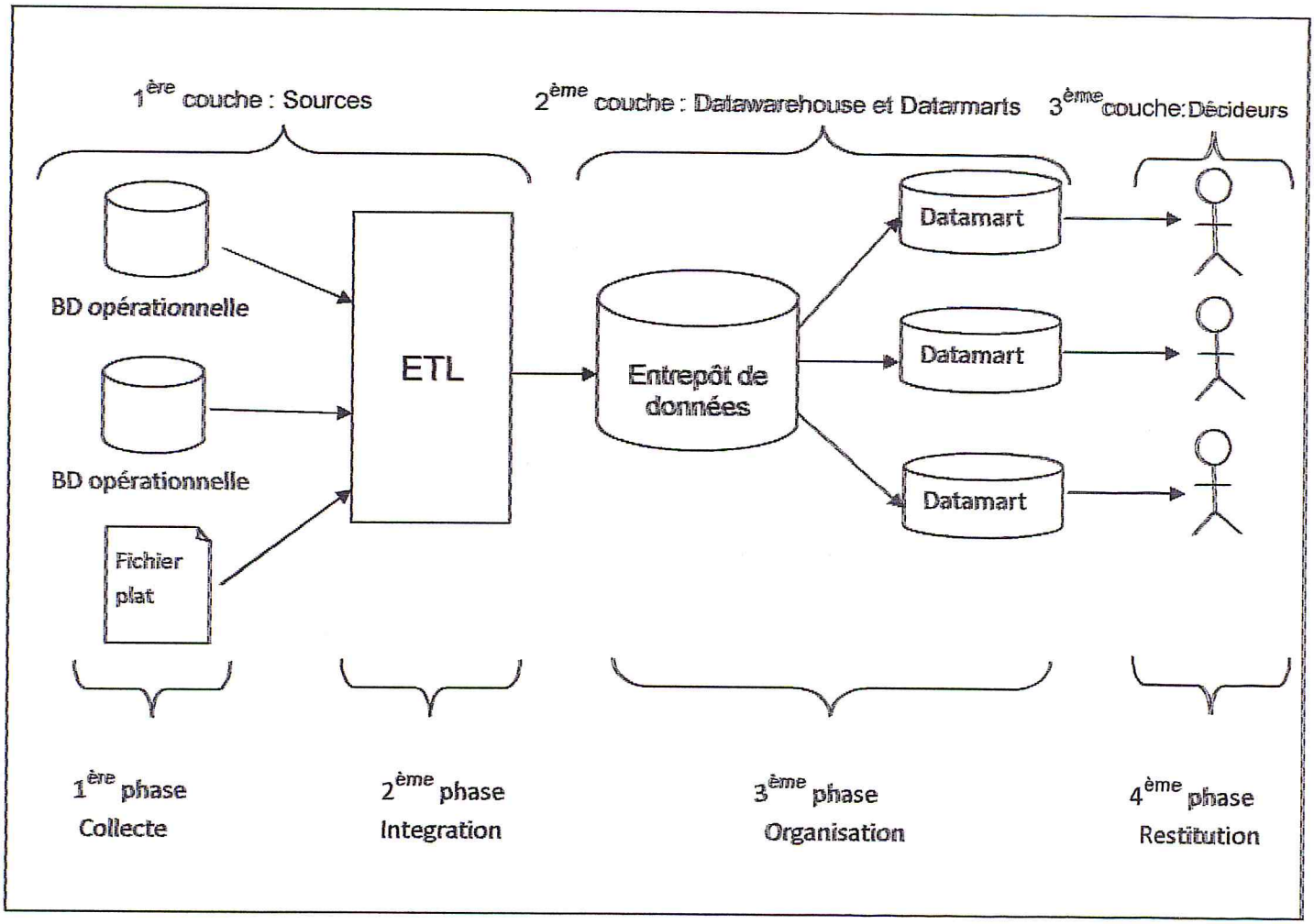

Figure 1 : L'architecture d'un système décisionnel

3- Les principales phases d'un système décisionnel

3-1- La phase de collecte

« La collecte s'effectue à partir de données appelées « **données sources** ». Ces données peuvent se présenter sous différents formats :

- Sources multiples hétérogènes (BD relationnelles, fichiers…)
- Autonomes, internes (BD production) ou externes (clients, fournisseurs…). » **[Kimball, 2001]**.

Vue la nature de ces sources de données il va falloir passer par une phase dite d'**intégration** pour pouvoir les manipuler avant de les stocker dans notre système d'aide à la décision.

3-2- La phase d'intégration

C'est à ce niveau qu'apparait la première couche logicielle de l'environnement décisionnel à savoir l'**ETL** (Extract-Transform-Load). « Cette couche offre des fonctions d'extraction de données issues de différents systèmes (internes ou externes), de transformation de ces données (homogénéisation, filtrage, calcul…) et de leur chargement dans un **datawarehouse** (DW ou entrepôt de données). Elle garantit la délocalisation de la charge de calcul et une meilleure disponibilité des sources. » **[kimball, 2004]**.

« La deuxième couche logicielle est l'**operational data store (ODS)** qui fait office de structure intermédiaire destinée à stocker les données issues des systèmes de production opérationnels. Ce sont en quelque sorte des zones de préparation avant l'intégration des données dans le datawarehouse : périodicité journalière, données qualifiées, premier niveau de filtrage et d'agrégat. En général, il existe deux types de schéma : un schéma « **ODS brut** » qui contient les tables qui reçoivent les données brutes des différentes sources et un schéma « **ODS final** » qui contient des tables avec une structure (champs et contraintes associées) le plus proche possible du schéma du datawarehouse (même si les tables de celui-ci peuvent contenir

plus de champs que les tables du datawarehouse) car ces données vont ensuite être figées dans l'entrepôt. L'ODS ne contient des données que sur une faible période et ces données vont être manipulées, transformées, traitées et modifiées plusieurs fois avant d'être copiées dans le DW.» **[Inmon, 2005]**.

Nous pouvons se passer de l'utilisation d'un ODS dans le cas où les données du DW sont une simple copie des données de production (sources) ce qui n'est malheureusement pratiquement jamais le cas dans de grosses structures.

3-3- La phase d'organisation

La troisième phase permet de stocker les données dans un entrepôt appelé « datawarehouse ». « L'entrepôt de données est une structure informatique dans laquelle est centralisé un volume important de données. » **[Imhoff,Galemmo,Geiger, 2003]**. Ces dernières sont extraites à partir d'une base de données sources, elles sont sélectionnées, transformées et chargées dans le DW.

L'entrepôt de données est rarement utilisé directement par les décideurs car :

- Contient plus que nécessaire pour une classe de décideurs.
- Structure relationnelle peu adaptée à l'analyse de données.

C'est à cause de ces problèmes que les **Magasins de données « Datamarts »** sont apparus.

« Le datamart est une extraction d'une partie d'un entrepôt de données dédiée à une fonction d'entreprise pour des raisons d'accessibilité, de facilité d'utilisation ou de performance. Les données sont généralement équivalentes à celles présentes dans le DW principal mais elles sont représentées de façon adaptée aux besoins spécifiques de la fonction et/ou du domaine de l'utilisateur (par exemple, nous allons créer un datamart pour le service marketing ou commercial). » **[Imhoff,Galemmo,Geiger, 2003]**.

Les datamarts sont aussi souvent utilisés lorsqu'une entreprise ne peut plus multiplier les optimisations sur son entrepôt de données sans pénaliser d'autres applications. Elle crée alors un nouvel environnement dédié à cette nouvelle application dont elle peut gérer librement les index.

3-4- La phase de restitution

La dernière phase concerne la restitution des résultats. Nous distinguons à ce niveau plusieurs types d'outils différents :

- Les outils de **reporting** et de **requêtes.**

- Les outils d'**analyse.**

- La phase de **Datamining.**

« Les outils de **reporting** et de **requêtes** permettent la mise à disposition de rapports périodiques. Ils offrent une couche d'abstraction orientée métier pour faciliter la création de rapports par les utilisateurs eux-mêmes en interrogeant le datawarehouse (l'entrepôt de données) grâce à des analyses croisées. Ils permettent également la production de tableaux de bord avec des indicateurs de haut niveau pour les managers, synthétisant différents critères de performance.

Les outils d'**analyse OLAP** permettent de traiter des données et de les afficher sous forme de cubes multidimensionnels et de naviguer dans les différentes dimensions. Cet agencement des données permet d'obtenir immédiatement plusieurs représentations d'un même résultat, en une seule requête sous une approche descendante des niveaux agrégés vers les niveaux détaillés (Drill-down, Roll-up).

Les outils de **Datamining** offrent une analyse plus poussée des données historisées permettant de découvrir des connaissances cachées dans les données comme la détection de corrélations et de tendances, l'établissement de typologies et de segmentations ou encore des prévisions. Le Datamining

est basé sur des algorithmes statistiques et mathématiques, et sur des hypothèses métier. » **[kimball, 2004]**.

4- L'entrepôt de données (Le datawarehouse)

4-1- Historique [Web 1]

Les entrepôts de données ont une longue histoire. Citons pour mémoire quelques grandes dates clés.

En 1958, le chercheur d'IBM Hans Peter Luhn a utilisé le terme Business Intelligence qu'il a défini comme la capacité de présenter les inter-relations entre des faits de telle sorte que cela permette de guider les actions pour atteindre le but espéré.

En 1992, Le concept du datawarehouse a été formalisé par Bill Inmon dans son ouvrage intitulé « Developing the Data Warehouse » qui nous montre comment construire l'entrepôt de données.

Un Datawarehouse est une collection de données thématiques, intégrées, non volatiles et historisées pour la prise de décisions.

En 1996, Ralph Kimball a publié son ouvrage « The Data Warehouse Toolkit ».

4-2- Définition d'un datawarehouse

Le datawarehouse constitue l'un des grands axes abordés dans l'informatique décisionnelle.

Le datawarehouse est défini de plusieurs façons comme suit :

- « Est une collection de données orientées sujet, intégrées, non volatiles et historisées, organisées pour le processus de décisions.» **[Inmon, 2002]**.
- « Désigne à la fois la base dans laquelle sont stockés l'ensemble des informations de production ainsi que l'ensemble du système d'information décisionnel.» **[Kimball,2001]**.

- « Fait référence à une collection de technologies d'aide à la décision permettant à des managers, dirigeants et analystes de prendre des décisions pertinentes et rapides. » **[Chaudhuri, 1997]**.

- Un datawarehouse est une vision centralisée et universelle de toutes les informations de l'entreprise. C'est une structure comme une base de données, qui à pour but de regrouper les données de l'entreprise pour des fins analytiques et pour aider à la décision stratégique contrairement aux bases de données.

La décision stratégique étant une action faite par les décideurs de l'entreprise, elle vise à améliorer, qualitativement ou quantitativement la performance de celle ci.

Si un entrepôt de données utilise le principe des bases de données relationnelles, il se distingue par de nombreux points.

Tout d'abord, il n'applique pas un modèle relationnel précis, car les tables n'ont pas toujours une structure commune. Les entrepôts de données servent à croiser des informations non liées directement (exemple : rattacher les informations des systèmes de production avec celles du support client pour en tirer des requêtes de qualité).

4-3- Les données dans le datawarehouse [Kimball, 2002]

Dans un datawarehouse, les données sont :

- Sélectionnées et préparées pour répondre aux questions importantes de l'entreprise.
- Intégrées à partir des différentes sources de renseignements.
- Datées ou possédant un numéro de version pour garder la trace de leur origine, éviter de recouvrir une information déjà présente dans la base de données et permettre de suivre l'évolution de cette information au cours du temps.

- Non volatiles, elles sont stables, en lecture uniquement et non modifiables. Pour conserver la traçabilité des informations et des décisions prises, les informations stockées au sein du datawarehouse ne doivent pas disparaître. Une requête lancée plusieurs fois à des mois d'intervalle doit restituer les mêmes résultats. Ainsi, dès lors qu'une donnée a été qualifiée pour être introduite au sein du datawarehouse, elle ne peut ni être altérée, ni modifiée, ni supprimée. Cette caractéristique diffère de la logique des systèmes de production qui bien souvent remettent à jour les données à chaque nouvelle transaction.

- Présentées selon différents axes d'analyse ou **dimensions** (par exemple : le temps, les types ou segments de clientèle, les différentes gammes de produits, les différents secteurs régionaux ou commerciaux, etc.). Le datawarehouse est conçu pour contenir les données en adéquation avec les besoins actuels et futurs de l'organisation, et répondre de manière centralisée à tous les utilisateurs. Ainsi, il n'y a pas de règle qui s'applique en matière de stockage, ni de modélisation unique. Le datawarehouse peut contenir certaines informations détaillées, issues des sources de production, nécessaires à un besoin de pilotage opérationnel récurrent, tout comme des **tables de faits**, prêtes à l'emploi.

- Orientées métiers ou business. Dans un datawarehouse, les données sont structurées par thèmes par opposition à celles organisées dans les systèmes de production par processus fonctionnel. L'intérêt de cette organisation est de disposer de l'ensemble des informations utiles sur un sujet. On peut ainsi passer d'une vision transversale de l'entreprise à une vision verticale.

4-4- Objectif d'un datawarehouse

Voici les objectifs de l'entrepôt de données tels que définis par Ralph Kimball, dans son ouvrage « *Entrepôt de données, Guide pratique du concepteur de Data Warehouse* » **[Kimball, 2001]**.

- L'entrepôt de données doit rendre les données de l'organisation facilement accessibles.
- L'entrepôt de données doit présenter l'information de l'organisation de manière cohérente.
- L'entrepôt de données doit être adaptable et résistant aux changements : les modifications de l'entrepôt de données doivent se faire en douceur, ce qui veut dire qu'elles ne doivent pas invalider les données existantes ou les applications.
- L'entrepôt de données doit être un bastion sûr protégeant notre richesse informationnelle.
- L'entrepôt de données doit être le socle sur lequel repose l'amélioration des prises de décision.

Pour permettre la prise de décision en utilisant les données de l'entreprise, il va falloir concevoir un datawarehouse dont le model conceptuel est un modèle multidimensionnel.

5- Modélisation multidimensionnelle

Pour mieux définir les concepts de la modélisation multidimensionnelle qui a été introduite par Ralph Kimball, nous devons répondre aux questions suivantes : Qu'est ce qu'un modèle ? Quelle est la différence entre les systèmes transactionnels et les systèmes décisionnels ?

« Un modèle est la représentation d'un objet, d'un système ou d'une idée sous une forme autre que celle de l'entité elle-même. Sa fonction est d'aider à expliquer, à comprendre ou à améliorer un système.

Le modèle de données est le cœur d'un système décisionnel. Toutes les expériences ont montré que la modélisation d'un système décisionnel nécessite des approches spécifiques différentes des approches utilisées dans les systèmes transactionnels. En effet, les techniques couramment utilisées pour modéliser les données ont initialement été conçues pour qu'elles s'adaptent à des problématiques qui n'ont pas lieu d'être dans le cadre de la mise en œuvre d'un système décisionnel.

L'une des différences importantes entre les systèmes transactionnels dits classiques et les systèmes décisionnels est l'organisation des données dans le système. Un modèle dimensionnel contient les mêmes informations qu'un modèle Entité/Association, mais présente les données dans un format symétrique plus approprié pour faire l'analyse de données ». **[Shanon,1975]**.

« La modélisation multidimensionnelle est une approche dédiée aux systèmes décisionnels. Elle part du principe que l'objectif majeur de ce type de système est l'analyse de données quantitatives (les faits) par rapport à des données qualifiantes (les dimensions). » **[Kimball,2002]**.

5-1- Les faits [kimball, 2002]

Dans un entrepôt de données, les faits sont numériques, exemple : montant en argent des ventes, le nombre d'unités vendues d'un produit, etc.

Une table de fait est une table qui contient l'ensemble des mesures correspondant aux informations de l'activité à analyser selon divers axes d'analyse.

5-2- Les dimensions [kimball, 2002]

Une dimension est une table qui contient les axes d'analyse selon lesquels nous voulons étudier des données observables (les faits) qui, soumises à une analyse multidimensionnelle, donnent aux utilisateurs des renseignements nécessaires à la prise de décision.

Un axe d'analyse peut être des clients ou des produits d'une entreprise, une période de temps, des activités menées au sein d'une société, etc.

5-3- Les différents schémas du model dimensionnel

Il existe trois types de base de modèle dimensionnel :

- Schéma en étoile (Star schema).
- Schéma en flocons de neige (Snowflake schema).
- Schéma en constellation de faits (Multi-star schema).

5-3-1- Schéma en étoile (Star schema)

C'est la structure de données la plus utilisée et la plus appropriée aux requêtes d'analyses. Elle est simple à créer, stable et intuitivement compréhensible par les utilisateurs finaux.

« Ce schéma consiste en une grande table de faits et un cercle d'autres tables de dimensions. » **[kimball, 2002]**. Quand il est illustré, il ressemble à une étoile, c'est d'ailleurs l'origine du terme «En étoile».

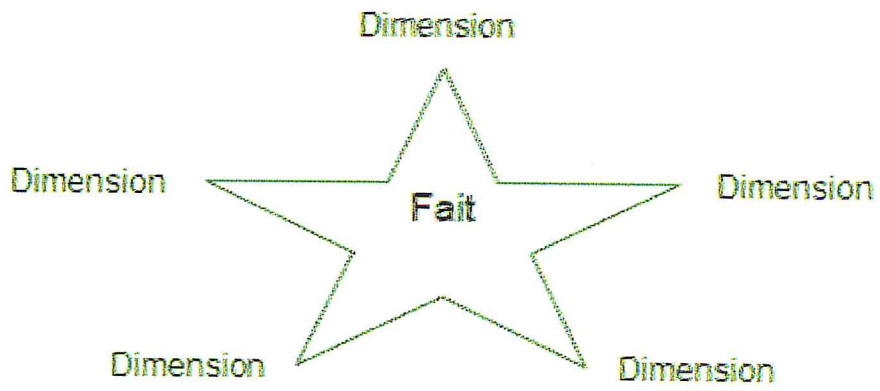

Figure 2 : Schéma en étoile

a- Avantages du schéma en étoile

- Lisibilité : Ce modèle est très parlant pour l'utilisateur car il présente de manière claire sa finalité. Il est orienté sujet et définit clairement les indicateurs d'analyse.
- Performance : Les chemins d'accès à la base sont prévisibles. Même si la table de fait comporte des millions de lignes, les tables de dimensions seront plus réduites.
- De plus, les tables de faits ne possèdent que des informations numériques et des identifiants.

b- Inconvénients du schéma en étoile

- Toutes les dimensions ne concernent pas les mesures.
- Redondance dans les dimensions.

5-3-2- Schéma en flocons de neige (Snowflake schema) [kimball, 2002]

Le schéma en flocons est une variante du schéma en étoile. Dans la théorie la différence réside dans la simple normalisation des tables de dimensions. Il est donc tout simplement question de mettre les attributs de chaque niveau hiérarchique dans une table de dimension à part.

a- Avantages du schéma en flocons de neige

- Normalisation des dimensions, réduisant la taille de chacune des tables.
- Formalisation de la notion de hiérarchie au sein d'une dimension (Drill down/Roll up).

b- Inconvénients du schéma en flocons de neige

- Plus complexe à gérer que les modèles en étoile.
- Navigation difficile due aux nombreuses jointures.

5-3-3- Schéma en constellation de faits (Multi-star schema)

« La modélisation en constellation consiste à fusionner plusieurs modèles en étoile qui peuvent utiliser des dimensions communes. » **[kimball, 2002]**. Donc ce modèle comprend plusieurs tables de faits et des tables de dimensions communes ou non à ces tables de faits.

6- Exemples des différents schémas du modèle dimensionnel

Pour l'illustration des schémas, nous utilisons la base de données **pubs** qui est fournie comme exemple dans le SGBD SQL Server de Microsoft. Cette base de données traite sur le domaine des publications d'ouvrages. Elle contient les auteurs, leurs ouvrages et les différents éditeurs ainsi que d'autres tables que le diagramme suivant montre :

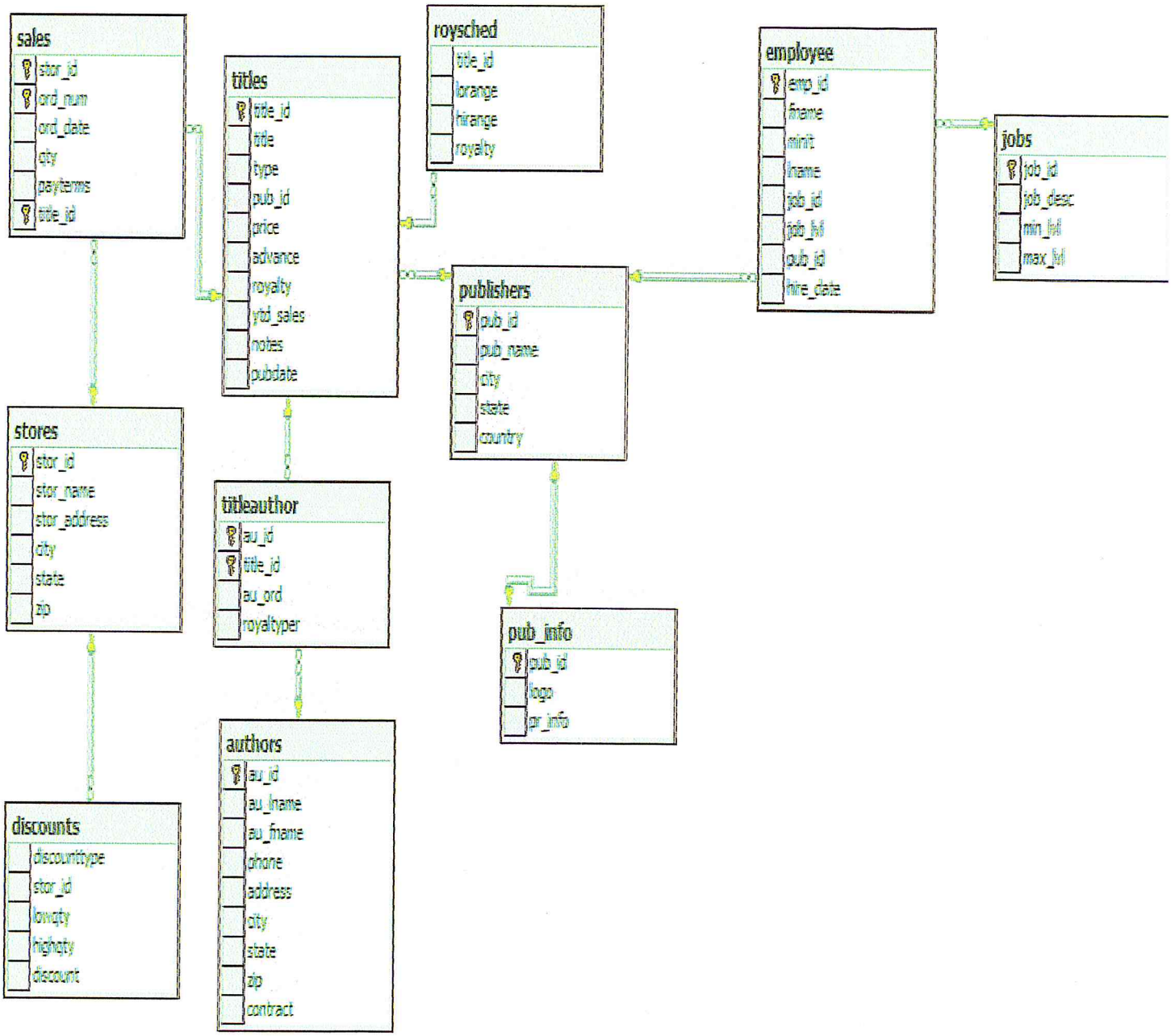

Diagramme 1 : Diagramme de la base de données pubs

6-1- Exemple du Schéma en étoile

Calcul du chiffre d'affaire (CA) des ventes des ouvrages selon les dimensions authors, titles, publishers, temps (date, mois, année) et stores. TF_CA est la table de fait qui contient les identifiants des tables de dimensions ainsi que la mesure CA.

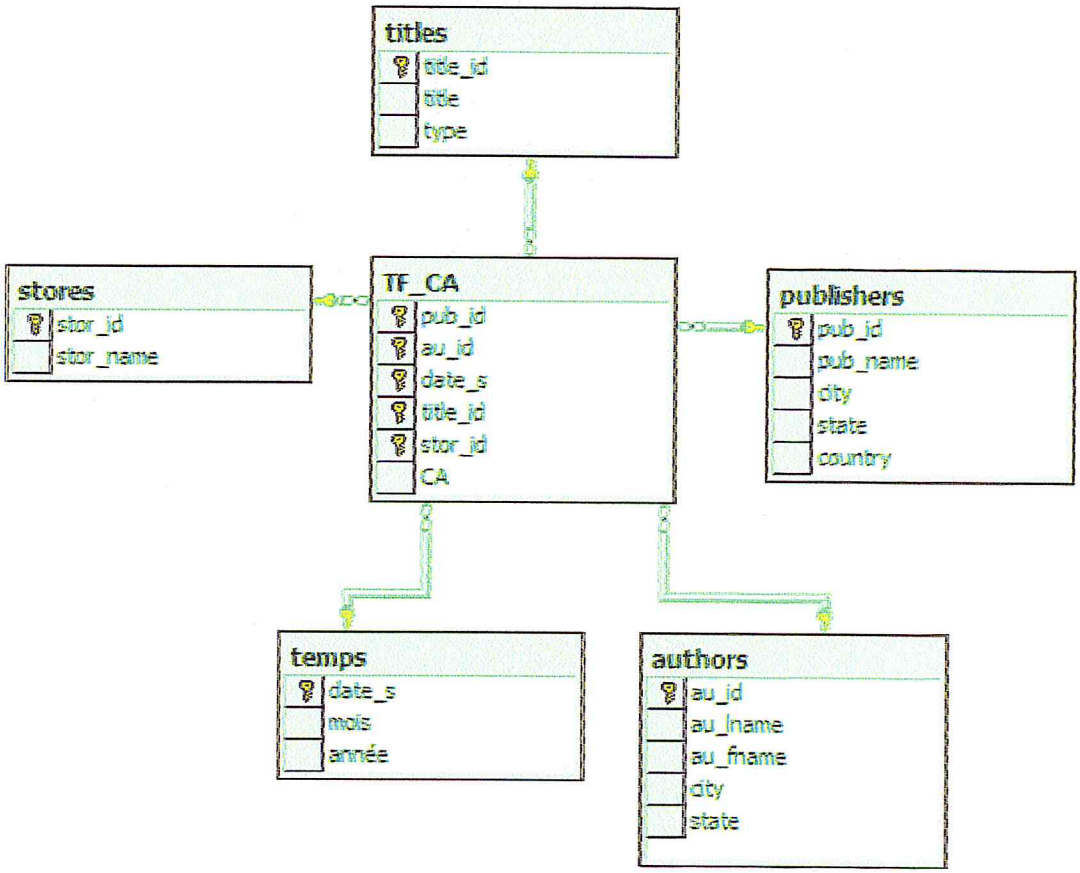

Diagramme 2 : Schéma en étoile

6-2- Exemple du Schéma en flocons de neige

Ce schéma normalise les tables de dimensions du schéma en étoile précédant en créant une hiérarchie des tables.

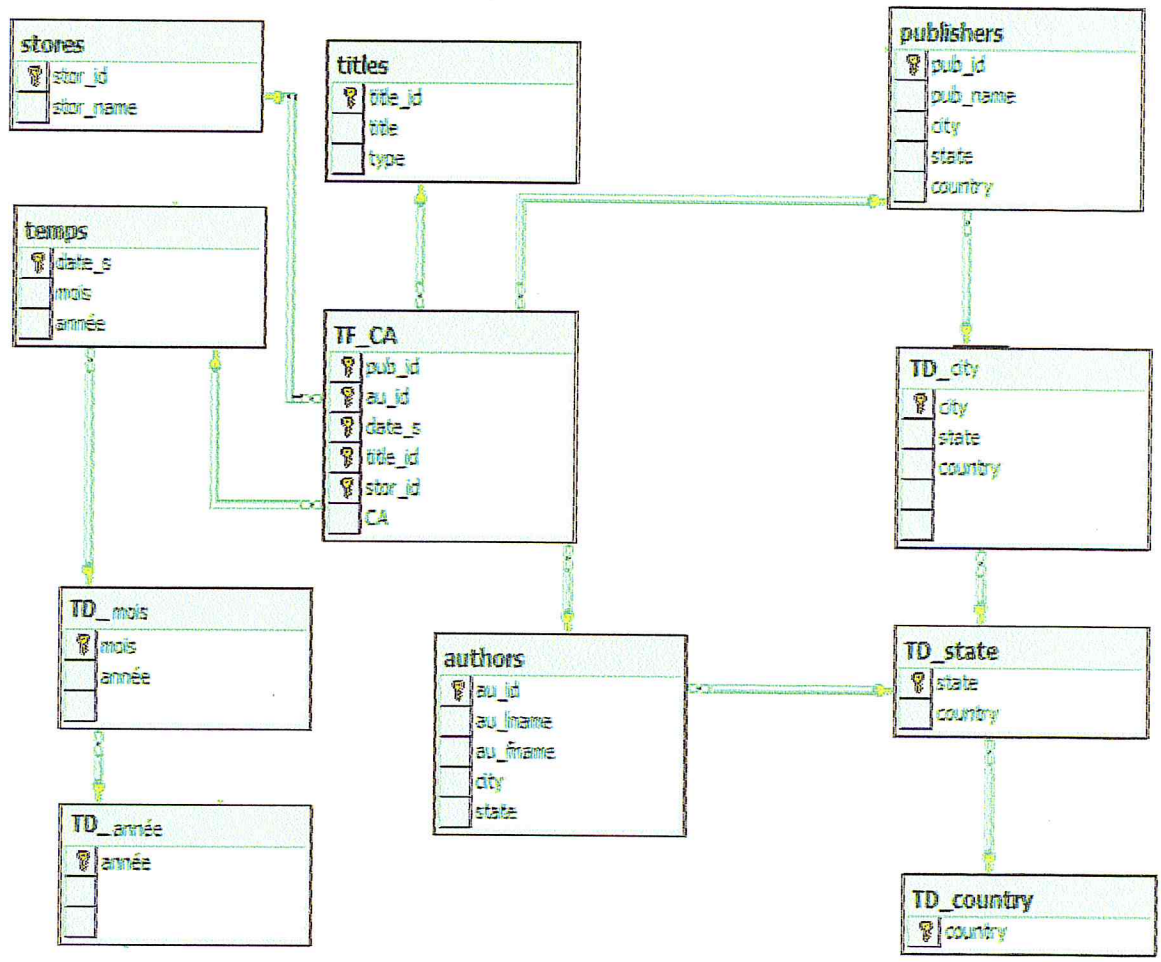

Diagramme 3 : Schéma en flocons de neige

Grâce à ces schémas, nous avons éclairci la signification des tables de faits et des tables de dimensions.

Avant de conclure ce chapitre, faisons une comparaison entre les systèmes transactionnels et les systèmes décisionnels.

7- Comparaison entre le système transactionnel et le système décisionnel

«Un système d'information est un ensemble organisé de ressources (matérielles, logicielles, personnelles, données, procédures...) permettant d'acquérir, de traiter, de stocker des informations (sous forme de données, textes, images, sons...) dans les organisations.

Il y'a deux systèmes d'information : transactionnel et décisionnel.

Le système d'information transactionnel : gère les applications quotidiennes et se rapproche à ce titre de la couche opérationnelle. Il est typiquement utilisé par les acteurs métiers afin de répondre à des besoins de simplification, d'automatisation et de gestion.

Le système d'information décisionnel : est utilisé pour aider à la prise de décisions de l'entreprise, et à ce titre doit permettre aux décideurs d'avoir un certain recul sur leur entreprise. Il fournit pour cela les informations nécessaires et pertinentes afin de faire les bons choix ». **[Reix, 2004]**.

Le tableau suivant résume les différences entre les systèmes décisionnels et transactionnels selon les données et l'usage des systèmes.

Différence	Systèmes transactionnels	Systèmes décisionnels
Par les données	Orienté applications métiers	Orienté thèmes et sujets (activités)
	Situation instantanée	Situation historique
	Donnée détaillées et codées non redondantes	Informations agrégées cohérentes souvent avec redondance
	Données changeantes constamment	Informations stables et synchronisées dans le temps
	Pas de référentiel commun	Un référentiel unique
L'usage	Assure l'activité au quotidien	Permet l'analyse et la prise de décision
	Pour les opérationnels	Pour les décideurs
	Mise à jour et requêtes simples	Lecture unique et requêtes complexes transparentes
	Temps de réponse immédiats	Temps de réponse moins critiques
	Faibles volumes à chaque transaction	Large volume manipulé
	Conçu pour la mise à jour	Conçu pour l'extraction
	Usage maîtrisé	Usage aléatoire

Tableau 1 : Tableau comparatif entre les systèmes transactionnels et les systèmes décisionnels. **[Akoka,Comyn-Wattiau, 2001].**

8- Conclusion

Le développement des premiers systèmes d'informations s'est concentré sur l'automatisation des processus opérationnels, le bon déroulement de l'activité principale de l'entreprise et l'excellence opérationnelle.

Les besoins d'analyse sont arrivés bien plus tard, ils relèvent plus de l'avantage concurrentiel et le pilotage de l'entreprise que de l'excellence opérationnelle.

Le système décisionnel cherche donc à donner un aperçu global de l'entreprise via des outils d'analyse pour aider les décideurs à prendre des décisions, et cela en organisant l'ensemble de données consolidées à partir des différentes sources dans une base de données unique 'Data Warehouse'. Pour l'alimentation de ce dernier, nous faisons appel au processus ETL que nous allons définir dans le chapitre suivant.

Chapitre II

Le processus ETL et le paradigme MapReduce

1- Introduction

Une base de données est un objet particulièrement difficile à définir puisqu'il est abordé en pratique selon différents points de vue :

- Pour un utilisateur, une base de données est un espace où il peut enregistrer des informations, les retrouver et les faire traiter automatiquement par un ordinateur.
- Pour un développeur une base de données est un ensemble de tables, de relations et de procédures écrites en SQL (Structured Query Language).
- Pour un administrateur informatique, une base de données est un ensemble de données à sauvegarder et à sécuriser.

Dans une base de données personnelle (que nous manipulons dans le logiciel Access de Microsoft par exemple), nous retrouvons essentiellement un schéma où nous sommes l'unique concepteur, développeur, fournisseur et analyste des données.

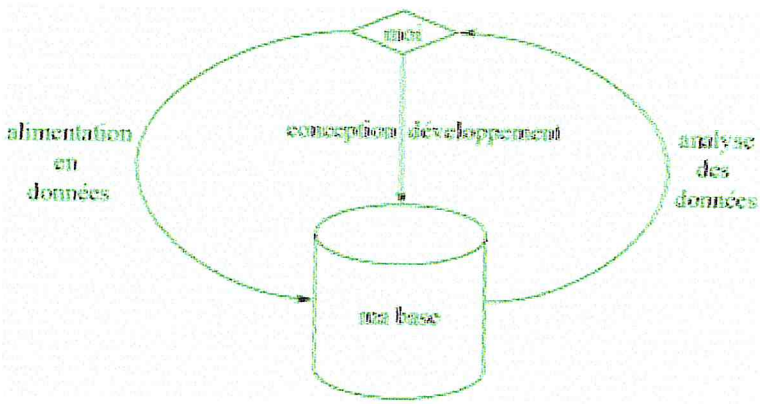

Figure 3 : Base de données personnelle **[Gruau,2004]**

Dans un SGBD professionnel (de type SQL Server, MySQL, Oracle,...) le schéma est fondamentalement différent. Les données sont fournies par plusieurs utilisateurs (parfois des milliers) à travers de multiples petites

transactions SQL. Ces données sont stockées dans une ou plusieurs bases de production et continuellement remises à jour par ces transactions. Cette partie du schéma constitue le système transactionnel.

L'autre partie du schéma constitue le système décisionnel où les données sont historisées dans un entrepôt dont l'élément constitutif n'est plus la table mais le cube. Ceci génère de gros transferts entre les deux systèmes mais les informations utiles sont plus proches des utilisateurs qui ont besoin d'analyser les données.

L'ensemble est géré, dans l'entreprise, par les concepteurs, les développeurs et les administrateurs du service informatique.

La figure suivante illustre les deux systèmes à savoir le transactionnel et le décisionnel :

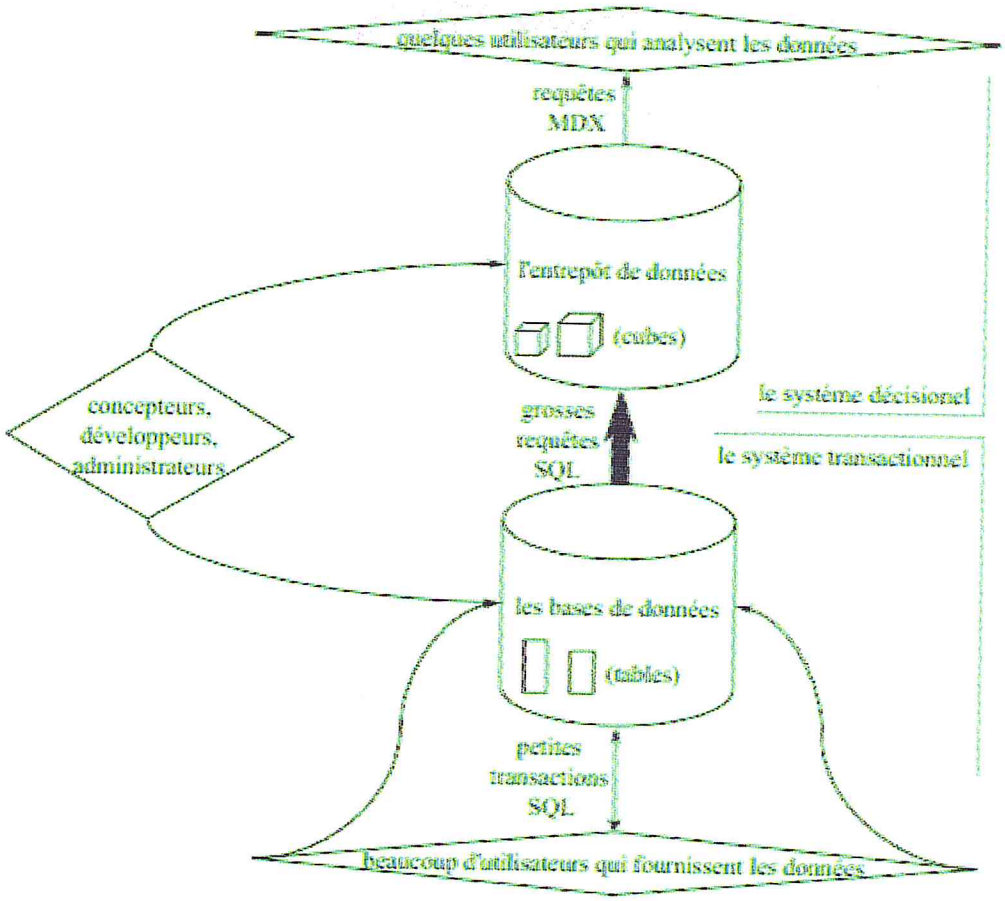

Figure 4 : Base de données professionnelle **[Gruau,2004]**

Comme exemple nous pouvons prendre n'importe quelle entreprise qui fabrique et vend des produits. Les utilisateurs qui fournissent les données sont : les vendeurs, les interlocuteurs auprès des fournisseurs et des usines. Les données sont stockées dans des tables représentant : les articles, les fournisseurs, les clients, les ventes et les stocks. Toutes ces informations seront regroupées sous forme de cubes concernant : les ventes par vendeur et par trimestre, la production par produit et par usine, etc. Dans cette entreprise, ces cubes sont susceptibles d'intéresser les managers du service commercial, du service marketing et du service logistique.

Nous représentons cet exemple par la figure suivante :

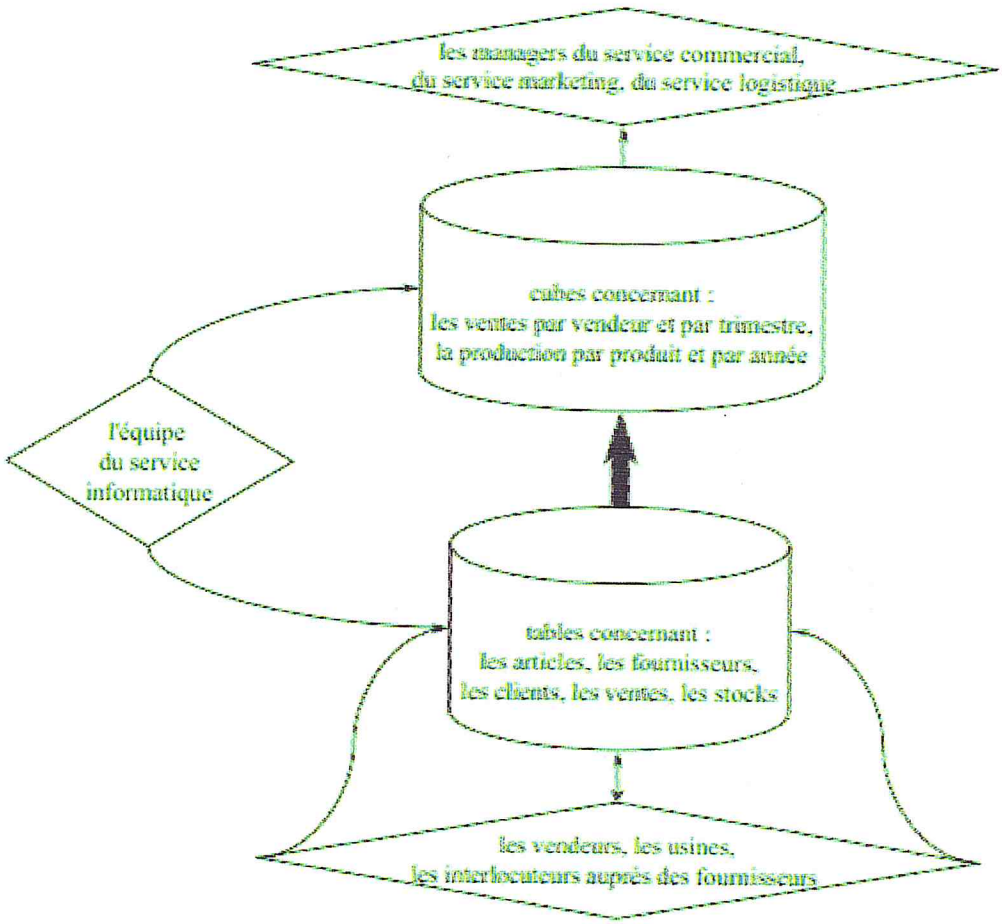

Figure 5 : Exemple de base de données professionnelle **[Gruau,2004]**

Le passage des bases de données à l'entrepôt de données est assuré par le processus ETL.

2- Définition de ETL

« ETL est le moyen qui permet l'alimentation du datawarehouse. Il stocke toutes les données qui proviennent des différents systèmes dans une seule grande base de données. » **[kimball, 2004]**. Nous parlons souvent d'outil et processus d'alimentation.

Le travail du processus ETL s'étend en trois étapes :

a- L'extraction des données.
b- La préparation/transformation des données
c- Le chargement des données dans l'entrepôt.

La figure suivante montre ces principales étapes :

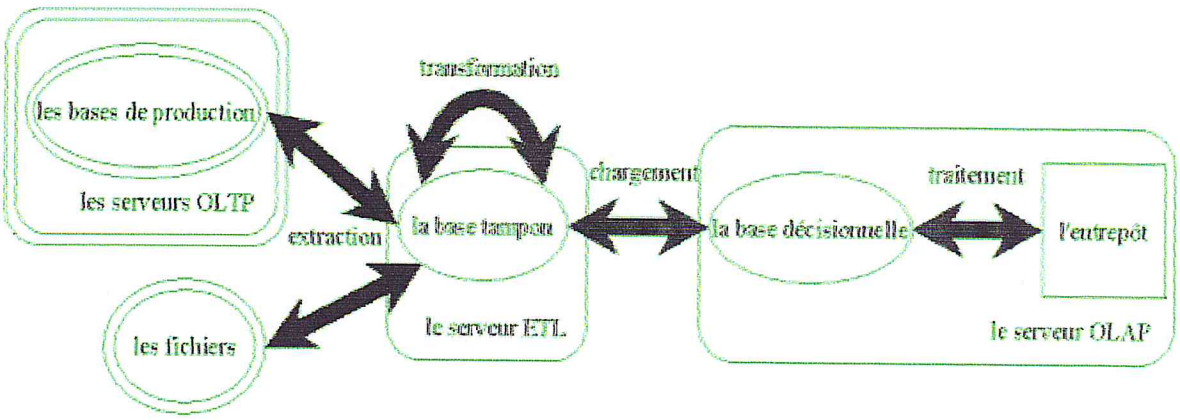

Figure 6 : Les étapes du processus ETL **[Gruau,2004]**

3- Les étapes du processus ETL

3-1- L'extraction des données

Selon Ralph Kimball, « L'extraction est la première étape du processus d'apport de données à l'entrepôt de données. Extraire, cela veut dire lire, interpréter les données sources et les copier dans la zone de préparation en vue de manipulations ultérieures.» **[kimball, 2004].**

ETL va lire sélectivement les données sources et les filtrer afin de n'extraire que l'information pertinente. L'extraction est périodique et se fait avec des règles ou des requêtes.

« Pour que l'étape d'extraction dure le moins longtemps possible, il faut que:

- La requête de sélection ne comporte aucune jointure (il faut donc extraire les tables une par une).
- Les données soient insérées dans des tables temporaires (elles n'ont aucune contrainte, aucun déclencheur et aucun index).

Par ailleurs, il est bon que dans les systèmes OLTP, chaque table concernée par l'extraction (clients, produits, etc.) soit munie d'une colonne pour la date de création et une autre pour la date de dernière modification. Sans ces colonnes, nous serons obligés d'extraire toutes les lignes et il serait compliqué de déterminer (dans la base tampon) les lignes réellement modifiées depuis la dernière extraction. Avec ces colonnes, l'extraction peut être incrémentale.

Dans tous les cas, le volume de données à extraire est important. Il y a toujours un choix à faire entre extraire toutes les lignes d'un coup (la méthode la plus rapide, mais comme cette transaction est non atomique, la moindre erreur est fatale à tout le processus) ou les extraire une par une (ce

qui prend plus de temps, mais permet de limiter l'effet d'une erreur ponctuelle). » **[Gruau,2004].**

3-2- La transformation et le contrôle des données

« Ce n'est pas parce que les données proviennent de bases de production qui fonctionnent rigoureusement bien que ces données sont valides pour le système décisionnel. Il faut presque toujours les transformer. » **[Gruau,2004].**

« La transformation des données est la fonctionnalité principale d'un ETL. Il doit disposer d'une fonction permettant de vérifier qu'une donnée est cohérente par rapport aux données déjà existantes dans la base centrale. Il doit aussi fournir des outils pour convertir des données différentes et doit être conçu pour manipuler de gros volumes de données. » **[kimball, 2004].**

Cette étape, qui du reste est très importante, assure en réalité plusieurs tâches qui garantissent la fiabilité des données et leurs qualités. Ces tâches sont :

- Consolidation des données.
- Correction des données et élimination de toute ambiguïté.
- Elimination des données redondantes.
- Compléter et renseigner les valeurs manquantes.

En effet, pendant que les données sont insérées dans les tables tampon, nous pouvons les uniformiser, c'est-à-dire les réparer, les compléter, les synchroniser et les formater.

Exemple de réparation des données : les codes postaux invalides peuvent être corrigés en utilisant un annuaire des codes postaux.

Exemple de complétion des données : déduire la région où est domicilié un propriétaire à partir du numéro d'immatriculation de son véhicule.

Rappelons que les bases de production n'utilisent pas forcement la même horloge, il faut donc synchroniser toutes les dates contenues dans les tables temporaires pendant leur transfert dans les tables tampon.

Par ailleurs, quand les données arrivent dans la base tampon, elles ne sont pas toutes au même format et ne respectent pas forcement le format de la base décisionnelle (généralement, les contraintes sur les chaînes de caractères ne sont pas les mêmes dans toutes les bases et les codages de date sont hétérogènes). Il faut donc uniformiser les formats avant le chargement : c'est le formatage.

3-3- Le chargement et le transfert des données

« C'est la dernière phase de l'alimentation d'un entrepôt de données, le chargement est une étape indispensable. Elle reste toute fois très délicate et exige une certaine connaissance des structures du système de gestion de la base de données (tables et index) afin d'optimiser au mieux le processus. » **[kimball, 2004]**.

Le chargement prend en compte la gestion du format final voulu des données. Comme les données sont chargées dans la base décisionnelle qui est munie d'un schéma, il faut en premier charger les tables qui ne contiennent aucune clé étrangère ensuite les tables qui ne contiennent que des clés étrangères vers des tables déjà chargées.

Pour la mise en œuvre du transfert de données, on distingue deux approches possibles :

- *Le transfert de fichiers* : ETL transporte les données du système source vers le système cible via un moteur.
- *Le transfert de base à base* : Dans ce cas, les outils travaillent en mode connecté, d'une source de données à une cible. Les données sont extraites de la source puis transférées à la cible en y appliquant éventuellement des transformations.

4- Les outils ETL

Les outils ETL, en français ETC « extraction-transformation-chargement » sont des outils qui garantissent la fiabilité et facilitent le déroulement des trois étapes citées précédemment. D'où leur importance dans un projet Data Warehouse.

Les programmeurs peuvent mettre en place les processus ETL en utilisant presque tous les langages de programmation, mais la construction de ces processus à partir de zéro peut devenir complexe, c'est pour cela que les entreprises achètent des outils ETL pour les aider à la création de processus ETL.

Le marché des outils ETL (qui sont très nombreux), est particulièrement morcelé. Il est malgré tout dominé par Informatica (PowerCenter).

Un bon outil ETL doit être en mesure de communiquer avec les différentes bases de données relationnelles et de lire les différents formats de fichiers utilisés dans une organisation.

Il existe plusieurs fournisseurs, citons:

- IBM Information Server-InfoSphere-DataStage.
- SAS Data Integration Studio.
- Oracle Warehouse Builder (OWB).
- Sap Business Objects Data Integration.

Les principaux outils ETL Open Source sont :

- Talend Open Studio.
- Pentaho Data Integration (PDI).
- Enhydra Octopus.
- Clover ETL.
- Ketl.

5- Performance de ETL

L'importance du processus ETL réside dans le fait qu'il est responsable d'alimenter plusieurs fois l'entrepôt de données. Comme les sources de données évoluent sans cesse suite aux transactions (Insert, Update et Delete), le processus est chargé de rafraichir l'entrepôt de données suite à ces événements transactionnels, ce qui fait de lui un processus continu.

« Les technologies traditionnelles de ETL rencontrent de nouveaux défis dus à l'accroissement des informations. De nos jours, l'entreprise recueille des centaines de gigaoctets de données pour le traitement et l'analyse.

La grande quantité de données rend ETL extrêmement lent en termes de temps. Pour l'évolution des environnements d'affaires, les utilisateurs ont une demande croissante d'obtenir des données dès que possible. L'utilisation de la parallélisation est la clé pour améliorer les performances et l'évolutivité de ces défis. » **[Liu,Thomson,Pederson, 2011]**.

Au cours de ces dernières années de nouvelles technologies ont vu le jour, comme le **cloud computing**[1] et le paradigme **MapReduce** qui ont été largement utilisés pour le calcul parallèle dans les zones de données à forte intensité.

6- Solution à la lenteur du processus ETL

Récemment, Liu, Thomson et Pederson (2011) ont présenté un prototype appelé **ETLMR** dans lequel ont été intégrés les concepts décisionnels, l'implémentation de stratégies de découpage et parallélisation des tâches ETL dans un environnement MapReduce. L'objectif de cette contribution est d'accélérer les performances d'un processus ETL. **[Liu,Thomson,Pederson, 2011]**.

7- Le paradigme MapReduce

MapReduce est un framework de développement informatique réalisé en C++, introduit par Google, dans lequel sont effectués des calculs parallèles et souvent distribués des données potentiellement très volumineuses. [Jaffre,Rauzy, 2010].

7-1- Pourquoi MapReduce

En parallèle à la multiplication exponentielle des informations disponibles, le prix des supports de stockage n'a cessé de baisser et la capacité d'augmenter. Les bases de données sont technologiquement prêtes à accueillir cette masse de données. Le téraoctet et ses multiples sont la nouvelle unité de mesure. Les principaux fournisseurs de bases de données pour data warehouse sont prêts notamment avec les solutions de virtualisation du stockage et de cloud computing pour l'entreprise.

Les data warehouses de dix ou cent téraoctets ne font d'ailleurs plus l'exception. Les sites internet de renommée mondiale comme Facebook gèrent des bases de données de l'ordre du petaoctet.

Encore faut-il disposer des capacités de traitement pour diriger, stocker, classer et traiter en un temps relativement raisonnable cette avalanche de données. C'est là qu'intervient le framework MapReduce.

7-2- Les étapes de MapReduce

Ce paradigme se présente en deux étapes :

7-2-1- L'étape Map

« Dans cette étape, le nœud à qui est soumis un problème, le découpe en sous-problèmes, et les délègue à d'autres nœuds (qui peuvent en faire de même récursivement). Les sous-problèmes sont ensuite traités par les différents nœuds à l'aide de la fonction *Map* qui à un couple (clé, valeur) associe un ensemble de nouveaux couples (clé, valeur). » [Jaffre,Rauzy, 2010].

Exemple de la fonction Map :

A un couple (UserId, User), on assigne le couple (Rôle, User). A l'issue de cette étape, on obtient une liste contenant les utilisateurs groupés par rôle.

L'étape du mapping est illustrée dans la figure suivante :

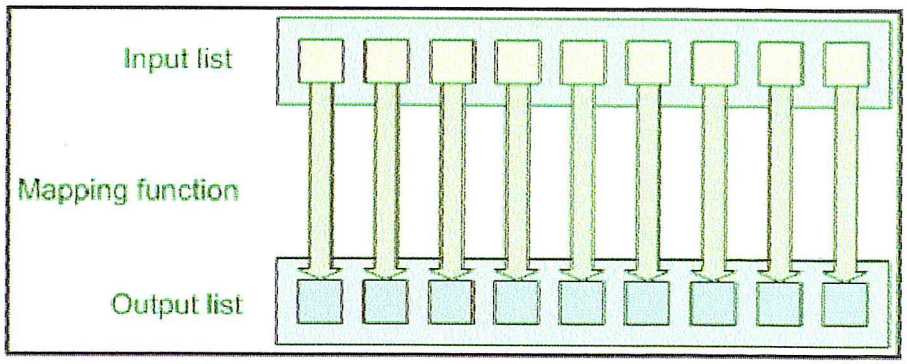

Figure 7 : L'étape Map de MapReduce **[Web 2]**

7-2-2- L'étape Reduce

« Dans cette étape, les nœuds les plus bas font remonter leurs résultats au nœud parent qui les a sollicité. Celui-ci calcule un résultat partiel à l'aide de la fonction *Reduce* (réduction) qui associe toutes les valeurs qui correspondent à la même clé à une unique paire (clé, valeur). Puis il remonte l'information à son tour.

À la fin du processus, le nœud d'origine peut recomposer une réponse au problème qui lui a été soumis. » **[Jaffre,Rauzy, 2010]**. La figure suivante montre l'étape du réducing :

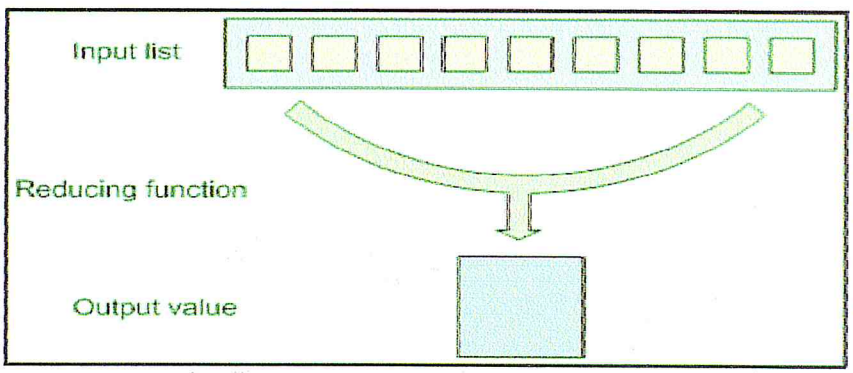

Figure 8 : L'étape Reduce de MapReduce **[Web 2]**

Remarque :

L'étape de mapping peut être parallélisée en traitant l'application sur différents nœuds du système pour chaque couple (clé, valeur).

L'étape de réducing n'est pas parallélisée et ne peut être exécutée avant la fin de l'étape de mapping.

La figure suivante montre la synchronisation des deux étapes :

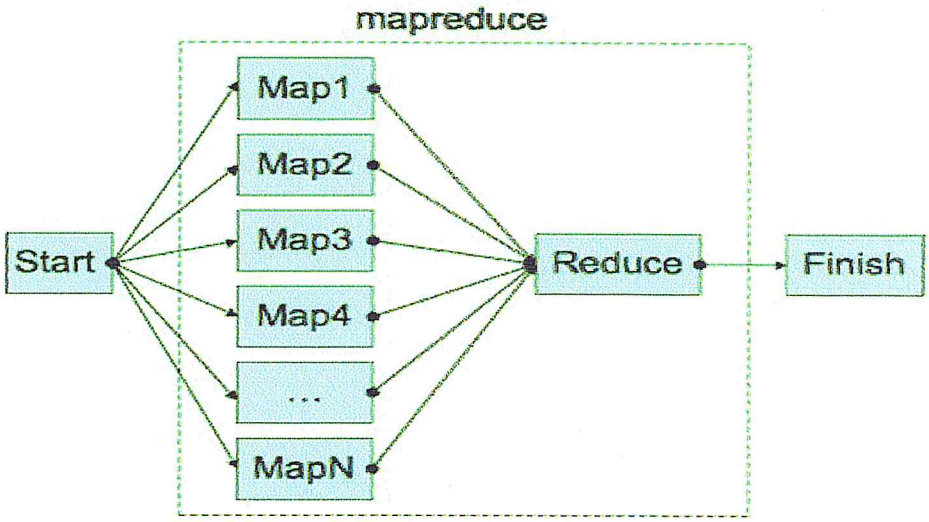

Figure 9 : Les étapes de MapReduce **[Web 2]**

7-3- Exemple de MapReduce

Pour illustrer l'algorithme MapReduce, considérons un jeu de données constitué des 3 phrases suivantes :

- savoir être et
- savoir faire
- sans faire savoir

Le but de l'illustration est d'appliquer le modèle *MapReduce* afin de compter le nombre d'occurrences des mots constituant le texte.

L'algotithme de MapReduce pour cet exemple est : **[Jaffre,Rauzy, 2010]**

```
map(String key, String value):
// key: document name
// value: document contents
for each word w in value:
EmitIntermediate(w, "1");

reduce(String key, Iterator values):
// key: a word
// values: a list of counts
int result = 0;
for each v in values:
result += ParseInt(v);
Emit(AsString(result));
```

La figure suivante schématise l'ensemble du processus :

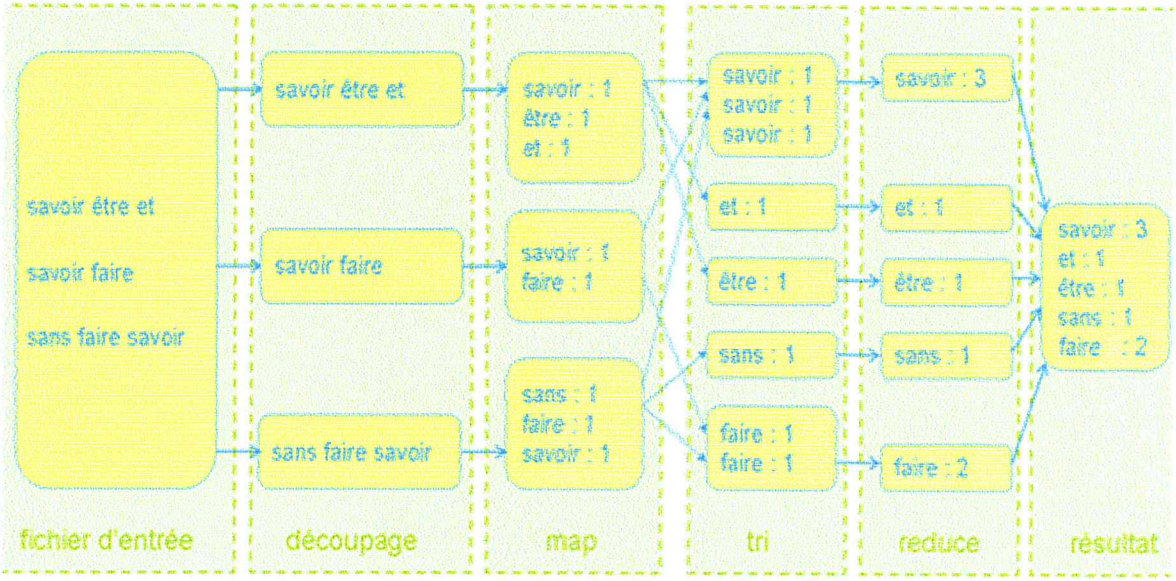

Figure 10 : Exemple du nombre d'occurrences d'un mot avec MapReduce **[Jaffre,Rauzy, 2010]**

7-4- Avantages et Inconvénients du MapReduce [Tannir, 2011]

- **Les Avantages:**
  - Traite de grands volumes de données.
  - Gère des milliers de processeurs.
  - Parallélise et distribue les traitements.
  - Ordonnance les entrées / sorties.
  - Gère la tolérance aux pannes.
  - Surveille les processus.
- **Les inconvénients:**
  - Une seule entrée pour les données.
  - Le flux de données en deux étapes le rend très rigide.
  - Le système de fichiers distribués (DFS) possède une bande passante limitée en entrée/sortie.
  - Les opérations de tris limitent les performances du Framework.

8- Conclusion

De nos jours la volumétrie des données pose un réel problème au sein des entreprises en termes de stockage, de temps d'accès et d'organisation. Ces données ne sont pas toutes de qualités. Pour cela, il existe un processus d'extraction, de transformation et de chargement des données importantes dans un entrepôt. Ce processus est ETL.

Dans ce chapitre, nous avons définit et montré l'importance du processus ETL dans un projet décisionnel. Cependant la quantité de données rend ETL extrêmement lent en termes de temps.

Comme solution à ce problème, nous avons présenté le paradigme MapReduce qui est une une technologie récente de la parallélisation et distribution des tâches pour améliorer les performances et l'évolutivité de ETL.

Il y a plusieurs implémentations de ce framework dans différents langages (C++, Java, Python, etc) et par de nombreux organismes (Google, Yahoo, etc). Les plus utilisés aujourd'hui sont : Le projet **Hadoop** et le projet **Disco.**

Dans le cadre de notre travail, nous nous intéressons à la mise en œuvre du processus ETL sous l'environnement Hadoop qui intègre le concept de MapReduce. Cet environnement sera présenté dans le chapitre suivant.

Chapitre III

L'environnement Hadoop Hive

1- Introduction

Dans ce chapitre, nous présentons l'un des projets qui implémente le paradigme MapReduce. Ce projet est le framework Hadoop dans lequel nous allons mettre en œuvre notre processus ETL.

2- Le projet Hadoop

« Après cinq ans de travail sur le moteur de recherche open source Nutch, *Doug Cutting* crée la plateforme qu'il baptise Hadoop, du nom de l'éléphant en peluche de son enfant. » **[White, 2009]**.

« En 2009, Hadoop fait partie des projets de la fondation logicielle Apache. Hadoop est un framework Java libre destiné aux applications distribuées et à la gestion intensive des données. Il permet aux applications de travailler avec des milliers de nœuds et des pétaoctets de données. Il dispose d'une implémentation complète de l'algorithme de MapReduce. » **[Web 3]**.

Plusieurs grands noms de l'informatique ont déclaré utiliser Hadoop, comme Facebook, Twitter, Yahoo et Microsoft.

« Aujourd'hui, Hadoop est une collection de sous-projets qui rentrent sous l'infrastructure du calcul distribué. Ces projets sont hébergés par Apache Software Fondation, qui fournit un support pour une communauté de projets open source. Bien que Hadoop est connu pour MapReduce et son système de fichiers distribués HDFS, les autres sous-projets offrent des services complémentaires. » **[White, 2009]**. Ces sous-projets sont illustrés dans la figure suivante :

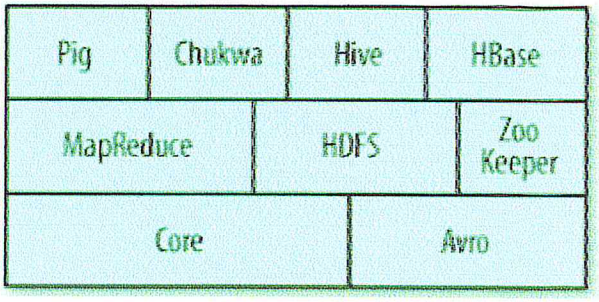

Figure 11 : Les sous-projets de Hadoop **[White, 2009]**

3- Les sous-projets de Hadoop

Les sous-projets de Hadoop sont :

3-1- Core

« Un ensemble de composants et d'interfaces pour le système de fichiers distribués et les entrées/sorties en général, (sérialisation, Java RPC, les structures de données persistantes). » **[White, 2009]**.

3-2- Avro

« Un système de sérialisation et stockage de données. » **[White, 2009]**.

3-3- MapReduce

« Un modèle de traitement de données distribué et un environnement d'exécution qui fonctionne sur un ensemble de machines. » **[White, 2009]**.

3-4- HDFS

« Un système de fichiers distribués qui fonctionne sur un ensemble de machines. » **[White, 2009]**.

3-5- Pig

« Un langage de flux de données et un environnement d'exécution pour explorer les ensembles de données très volumineux. Pig fonctionne avec HDFS et MapReduce. » **[White, 2009]**.

3-6- HBase

« Une base de données distribuée. HBase utilise HDFS pour le stockage, et le MapReduce pour le calcul. » **[White, 2009]**.

3-7- ZooKeeper

« Un service de coordination performant pour les applications distribuées. » **[White, 2009]**.

3-8- Hive

« Une infrastructure data warehouse. Hive stocke les données dans HDFS et fournie un langage de requêtes basé sur SQL qui s'exécutent en MapReduce. » **[White, 2009]**.

3-9- Chukwa

« Une collection de données distribué et système d'analyse. Chukwa exécute les collections qui stocke les données dans HDFS et utilise MapReduce pour fournir des rapports.» **[White, 2009]**.

Dans notre projet, nous nous intéressons aux sous-projets : MapReduce, HDFS et Hive.

4- MapReduce dans Hadoop

Dans le chapitre précédant, nous avons défini le paradigme MapReduce. L'environnement Hadoop dispose d'une implémentation complète de ce paradigme. A sa lancée, Hadoop génère des machines virtuelles (des nœuds) qui traitent les différentes sous tâches. (voire Figure 12).

```
hduser@ubuntu:/usr/local/hadoop$ bin/start-all.sh
starting namenode, logging to /usr/local/hadoop/bin/../logs/hadoop-hduser-namenode-ubuntu.
localhost: starting datanode, logging to /usr/local/hadoop/bin/../logs/hadoop-hduser-datan
localhost: starting secondarynamenode, logging to /usr/local/hadoop/bin/../logs/hadoop-hdu
starting jobtracker, logging to /usr/local/hadoop/bin/../logs/hadoop-hduser-jobtracker-ubu
localhost: starting tasktracker, logging to /usr/local/hadoop/bin/../logs/hadoop-hduser-ta
hduser@ubuntu:/usr/local/hadoop$
```

Figure 12 : Résultat de la lancée de Hadoop **[Noll,2007]**

Au niveau des deux dernières lignes que comporte la figure 12, nous remarquons la lancée de **jobtracker** et **tasktracker**.

- jobtracker : c'est le maître à qui une tâche est soumise.
- tasktracker : c'est les esclaves qui exécutent les sous tâches de la tâche soumise au jobtracker.

L'exécution d'une tâche sous Hadoop suit l'algorithme MapReduce, c'est-à-dire, elle passe par l'étape *Map* ensuite l'étape *Reduce*, comme nous l'avons montré dans le chapitre précédant. La figure suivante montre l'exécution de l'exemple wordcount sous Hadoop :

```
hduser@ubuntu:/usr/local/hadoop$ bin/hadoop jar hadoop*examples*.jar wordcount /user/hduser
10/05/08 17:43:00 INFO input.FileInputFormat: Total input paths to process : 3
10/05/08 17:43:01 INFO mapred.JobClient: Running job: job_201005081732_0001
10/05/08 17:43:02 INFO mapred.JobClient:  map 0% reduce 0%
10/05/08 17:43:14 INFO mapred.JobClient:  map 66% reduce 0%
10/05/08 17:43:17 INFO mapred.JobClient:  map 100% reduce 0%
10/05/08 17:43:26 INFO mapred.JobClient:  map 100% reduce 100%
10/05/08 17:43:28 INFO mapred.JobClient: Job complete: job_201005081732_0001
10/05/08 17:43:28 INFO mapred.JobClient: Counters: 17
10/05/08 17:43:28 INFO mapred.JobClient:   Job Counters
10/05/08 17:43:28 INFO mapred.JobClient:     Launched reduce tasks=1
10/05/08 17:43:28 INFO mapred.JobClient:     Launched map tasks=3
10/05/08 17:43:28 INFO mapred.JobClient:     Data-local map tasks=3
10/05/08 17:43:28 INFO mapred.JobClient:   FileSystemCounters
10/05/08 17:43:28 INFO mapred.JobClient:     FILE_BYTES_READ=2214026
10/05/08 17:43:28 INFO mapred.JobClient:     HDFS_BYTES_READ=3639512
10/05/08 17:43:28 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=3687918
10/05/08 17:43:28 INFO mapred.JobClient:     HDFS_BYTES_WRITTEN=880330
10/05/08 17:43:28 INFO mapred.JobClient:   Map-Reduce Framework
10/05/08 17:43:28 INFO mapred.JobClient:     Reduce input groups=82290
10/05/08 17:43:28 INFO mapred.JobClient:     Combine output records=102286
10/05/08 17:43:28 INFO mapred.JobClient:     Map input records=77934
10/05/08 17:43:28 INFO mapred.JobClient:     Reduce shuffle bytes=1473796
10/05/08 17:43:28 INFO mapred.JobClient:     Reduce output records=82290
10/05/08 17:43:28 INFO mapred.JobClient:     Spilled Records=255874
10/05/08 17:43:28 INFO mapred.JobClient:     Map output bytes=6076267
10/05/08 17:43:28 INFO mapred.JobClient:     Combine input records=629187
10/05/08 17:43:28 INFO mapred.JobClient:     Map output records=629187
10/05/08 17:43:28 INFO mapred.JobClient:     Reduce input records=102286
```

Figure 13 : Exécution de l'exemple wordcount sous Hadoop **[Noll,2007]**

Cette figure illustre la synchronisation des étapes de MapReduce, nous remarquons que l'étape *Reduce* ne commence que lorsque l'étape *Map* se termine (100%).

L'exécution de cet exemple a nécessité trois Map et un seul Reduce (Launched map task=3, Launched reduce task=1).

5- Hadoop Distributed File System (HDFS)

Quand un ensemble de données devient trop grand pour la capacité de stockage physique d'une seule machine, il devient nécessaire de le partitionner sur un certain nombre de machines séparées. Les systèmes de fichiers qui gèrent le stockage à travers un réseau de machines sont appelés **systèmes de fichiers distribués**. Étant donné qu'ils sont basés sur le réseau, cela les rend plus complexes que les systèmes de fichiers réguliers. Par exemple, l'un des plus grands défis est de rendre le système de fichiers capable de tolérer l'échec d'un nœud sans avoir à subir la perte de données.

Hadoop est livré avec un système de fichiers distribué appelé **HDFS** qui signifie **Hadoop Distributed File System**.

« HDFS est un système de fichiers conçu pour stocker des fichiers très volumineux en cours d'exécution sur un ensemble de machines. » **[White, 2009]**.

« HDFS a une architecture maître/esclave. Un cluster HDFS est constitué d'un seul **NameNode**, un serveur maître qui gère l'espace de noms du système de fichiers et réglemente l'accès aux fichiers par les clients. En outre, il y a un certain nombre de **DataNodes**, généralement un par nœud du cluster, qui gère le stockage attaché aux nœuds sur lesquels il s'exécute.

HDFS expose un système de fichiers espace de noms '**namespace**' et permet aux utilisateurs de données de les mémoriser dans les fichiers.

En interne, un fichier est divisé en un ou plusieurs blocs et les **blocs** sont stockés dans un ensemble de DataNodes. Le NameNode exécute des opérations du système de fichiers d'espace de nom comme l'ouverture, la fermeture, et le renommage de fichiers et de répertoires. Il détermine également le mappage des blocs aux DataNodes. Ces derniers, sont chargés de lire et écrire les requêtes des clients du système de fichiers. Aussi, ils effectuent la création de blocs, la suppression et la réplication sur instruction du NameNode.

Le NameNode et DataNodes sont des morceaux de logiciels conçus pour fonctionner sur des machines qui fonctionnent généralement avec un système d'exploitation GNU / Linux (OS).

HDFS est construit en utilisant le langage Java, n'importe quelle machine qui supporte Java peut exécuter le NameNode et le DataNode. » **[Web 4]**.

L'architecture de HDFS est illustrée dans la figure suivante :

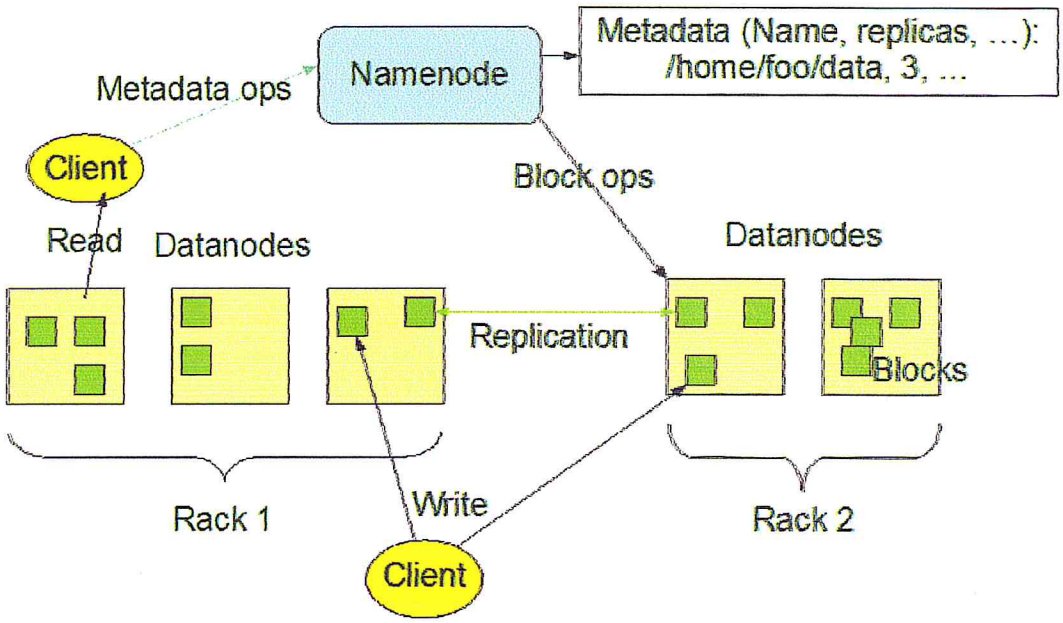

Figure 14 : L'architecture de HDFS **[Web 4]**

La lancée de Hadoop (figure 12) nous montre le lancement de NameNode et DataNode.

La figure suivante montre l'utilisation de MapReduce et HDFS pour le traitement des tâches dans le cluster Hadoop :

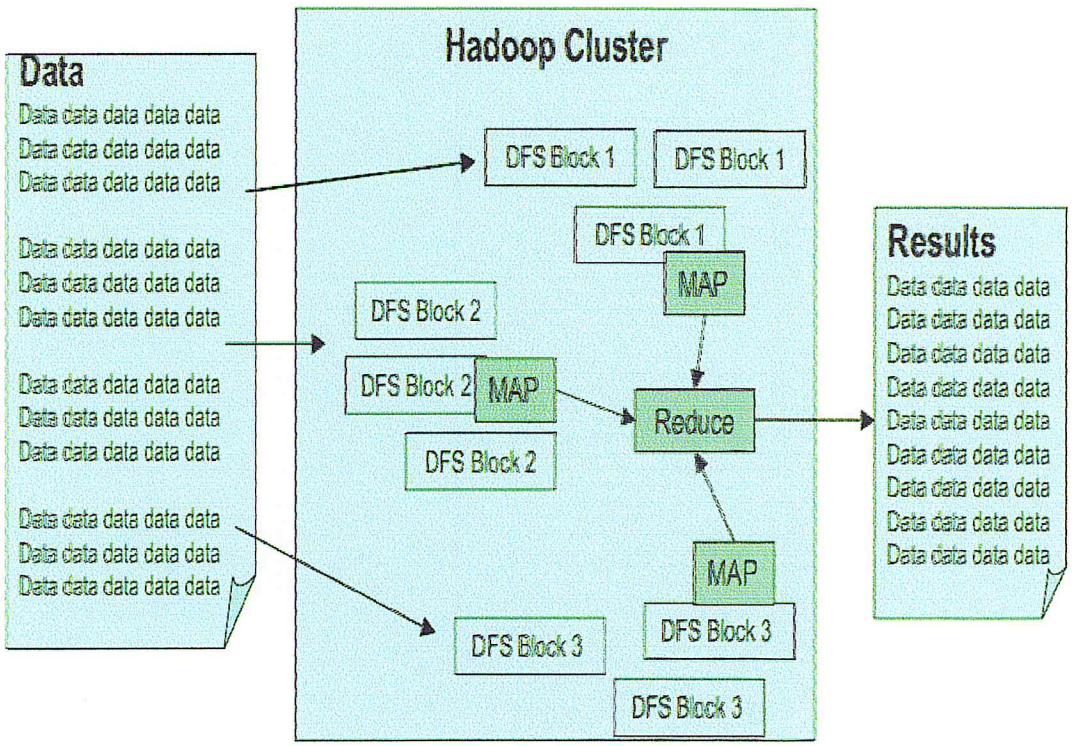

Figure 15 : Traitement des tâches dans le cluster Hadoop **[Sanjay,2008]**

6- Hive

« Hive est un logiciel d'entrepôt de données qui facilite l'interrogation et la gestion de grands ensembles de données résidant dans le stockage distribué. » **[Web 5]**. Hive a été initialement développé par Facebook.

« Hive fournit :

- Des outils pour faciliter l'extraction, la transformation et le chargement des données (ETL).
- Un mécanisme imposant une structure sur une variété de formats de données.
- L'accès aux fichiers stockés directement dans HDFS ou dans d'autres systèmes de stockage de données tels que HBase.
- Exécution des requêtes en MapReduce.

Hive définit un langage de requête proche du SQL, appelé HiveQL, qui permet aux utilisateurs familiers avec SQL d'interroger les données. Ce langage permet également aux programmeurs qui sont familiers avec le framework MapReduce d'utiliser leur mappers et reducers personnalisés pour effectuer des analyses plus sophistiquées qui ne peuvent être pris en charge par les fonctions intégrées de ce langage.

HiveQL peut également être étendu avec des fonctions scalaires personnalisées (UDF), agrégations (UDAF), et les fonctions de table (UDTF). » **[Web 5]**.

7- Hadoop Web Interfaces

Hadoop est livré avec plusieurs interfaces web qui fournissent des informations sur ce qui se passe dans le cluster Hadoop.

7-1- MapReduce jobtracker web interface

Jobtracker web interface fournit des informations sur les statistiques des jobs (les tâches) du cluster Hadoop qui sont en cours d'exécution, terminés ou échoués. Elle fournit aussi un fichier qui comprend l'historique des jobs et permet d'accéder à la machine locale sur laquelle l'interface est exécutée.

Par défaut, cette interface est disponible sur http://localhost:50030/

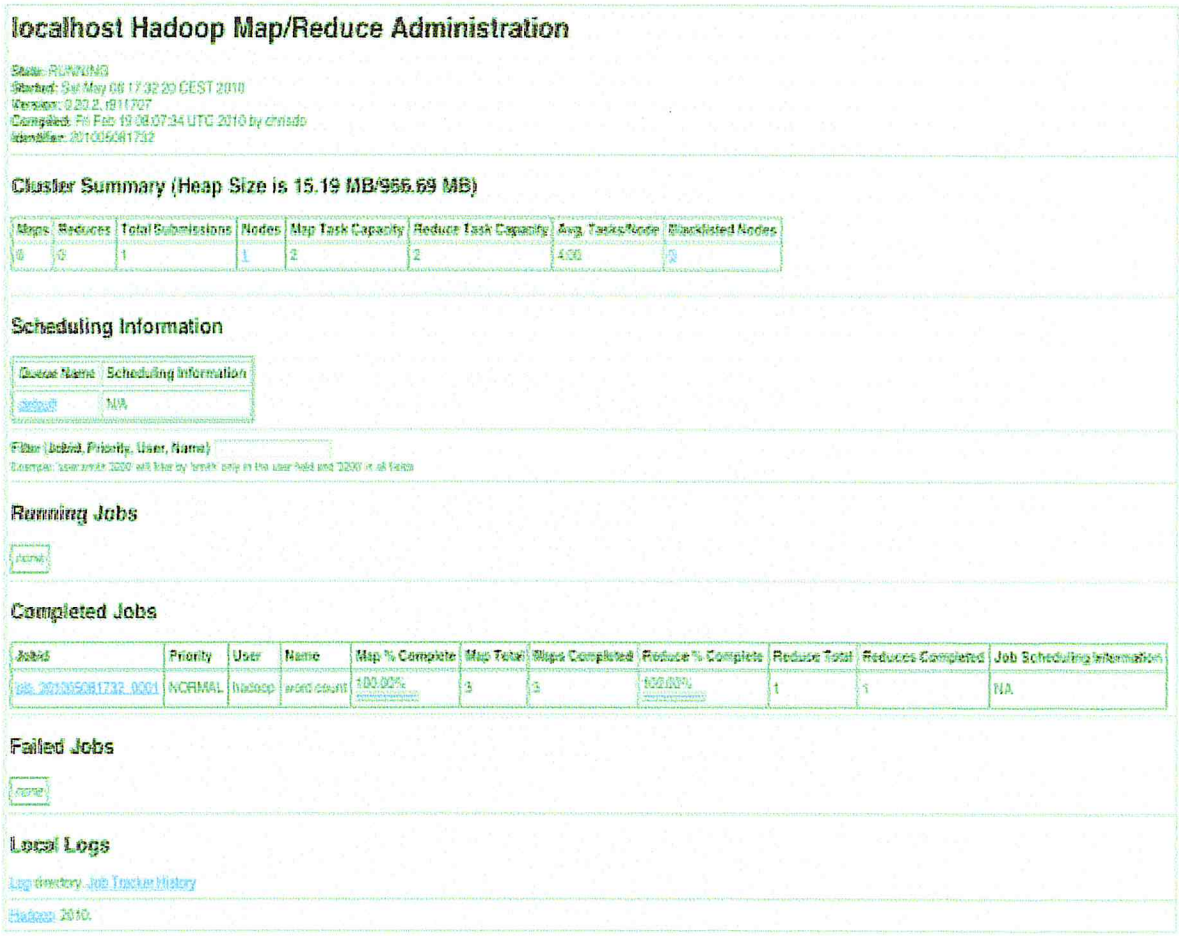

Figure 16 : MapReduce jobtracker web interface **[Noll, 2007]**

7-2- tasktracker web interface

L'interface web tasktracker montre les tâches en cours d'exécution / non en exécution. Il donne également l'accès à la machine locale.

Par défaut, cette interface est disponible sur http://localhost:50060/

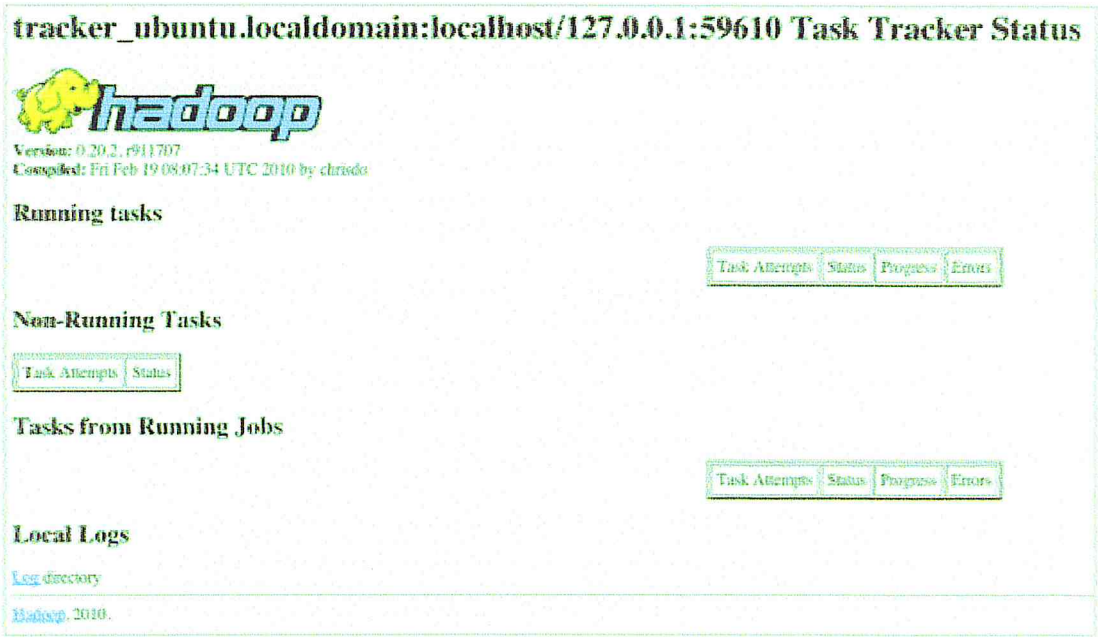

Figure 17 : tasktracker web interface **[Noll, 2007]**

7-3- HDFS NameNode web interface

L'interface NameNode montre un résumé du cluster, y compris des informations sur la capacité totale / restante, les nœuds vivants et morts. En outre, elle permet de parcourir HDFS namespace et afficher le contenu de ses fichiers dans le navigateur web. Elle donne également l'accès à la machine locale.

Par défaut, cette interface est disponible sur http://localhost:50070/

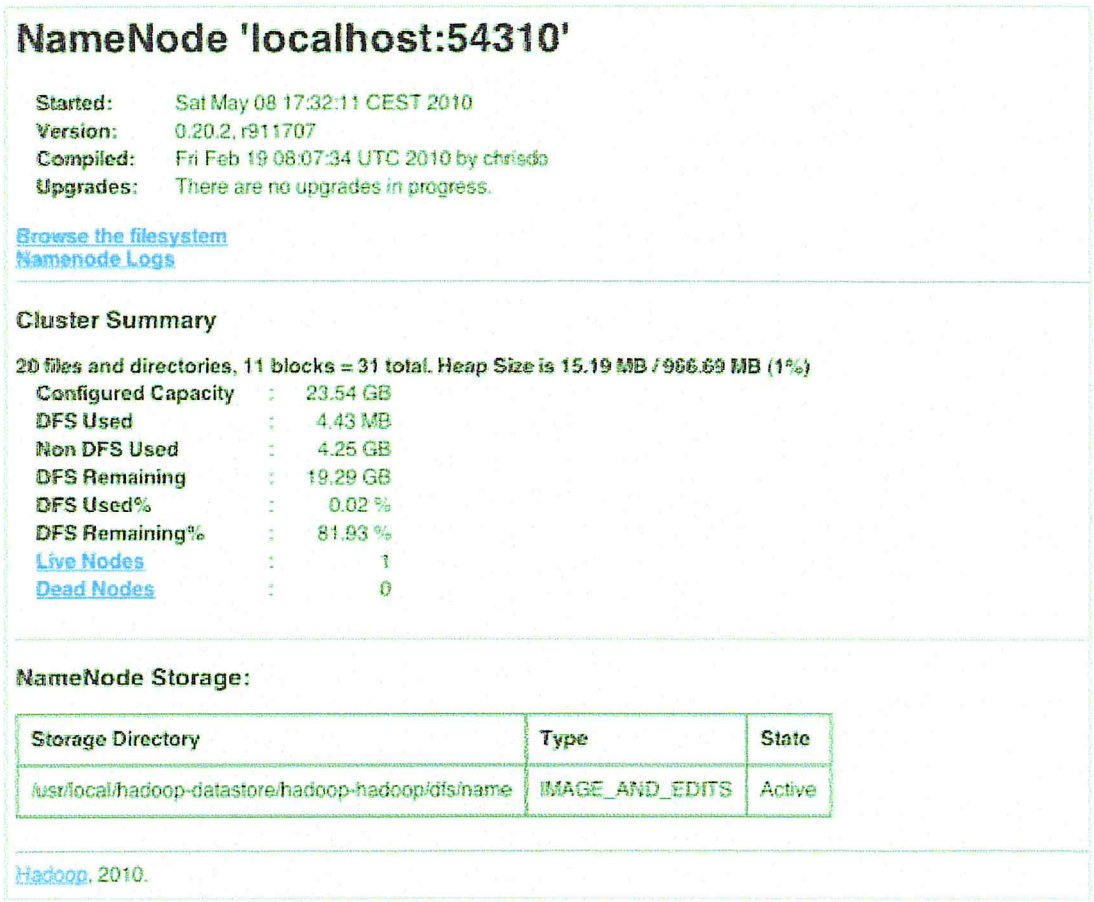

Figure 18 : NameNode web interface **[Noll, 2007]**

8- Conclusion

Dans ce chapitre nous avons défini le projet Hadoop, ses sous projets et les sous projets dont nous avons besoins pour la mise en œuvre du processus ETL.

Nous présentons dans le chapitre suivant la mise en œuvre du processus ETL sous l'environnement Hadoop Hive.

Chapitre IV

La mise en œuvre du processus ETL sous l'environnement Hadoop Hive

1- Introduction

En Août 2011, les chercheurs Liu, Thomsen et Pederson ont présenté le prototype ETLMR sous l'environnement Disco, c'est un ETL framework basé sur MapReduce comme le montre la figure suivante :

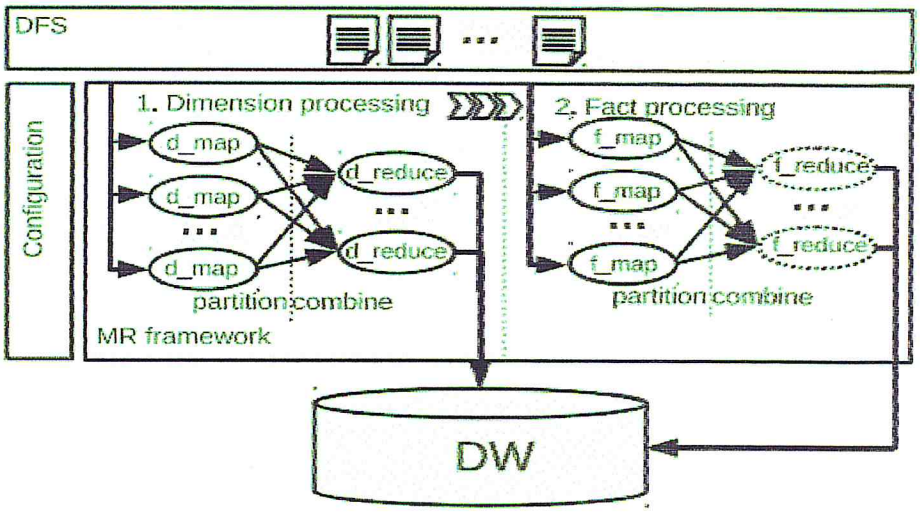

Figure 19 : ETLMR **[Liu,Thomson,Pederson, 2011]**

Contrairement aux chercheurs qui ont appliqué le MapReduce seulement pour le traitement (transformations) des tables de dimensions et tables de faits, nous allons mettre en œuvre le processus ETL sous l'environnement Hadoop Hive en appliquant le MapReduce sur les trois phases de ETL à savoir l'extraction, la transformation et le chargement des données.

2- Mise en œuvre du processus ETL

Pour mettre en œuvre notre processus ETL, nous suivons l'architecture illustrée dans la figure suivante :

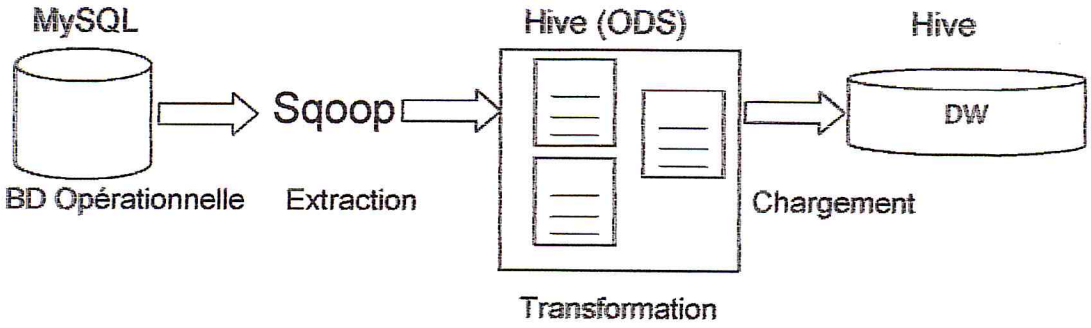

Figure 20 : Notre Architecture pour la mise en œuvre du processus ETL

Cette architecture est définie et expliquée dans les étapes suivantes :

Etape 1 : Cette étape présente l'installation du système.

1- Installation d'une machine virtuelle VMware player.
2- Installation de ubuntu Lucid 10.04 64 bits sur la VMware Player.
3- Installation du paquet cloudera CDH3U4 sur la VMware Player.
4- Installation et configuration de Hadoop sur cloudera.
5- Installation et configuration de Hive sur cloudera.
6- Installation et configuration de Sqoop sur cloudera.
7- Installation et configuration de MySQL sur cloudera.

Cloudera est une société qui se consacre au développement de logiciels fondés sur Apache Hadoop, permettant l'exploitation de **Big Data**, à savoir des bases de données accumulant plusieurs pétaoctets. Cloudera a été fondée en 2008 par le mathématicien Jeff Hammerbach, un ancien de Facebook, où il était en charge de l'analyse de données et du développement de programmes permettant un meilleur ciblage publicitaire.

La figure suivante montre le résultat de l'étape 1 :

Figure 21 : Cloudera vm

Etape 2: Une fois l'installation terminée, nous passons à la création de la base de données sur MySQL. Pour cela, nous prenons un extrait de la base de données pubs qui se trouve sur SQL server. Le résultat de cette étape est illustré dans les figures suivantes :

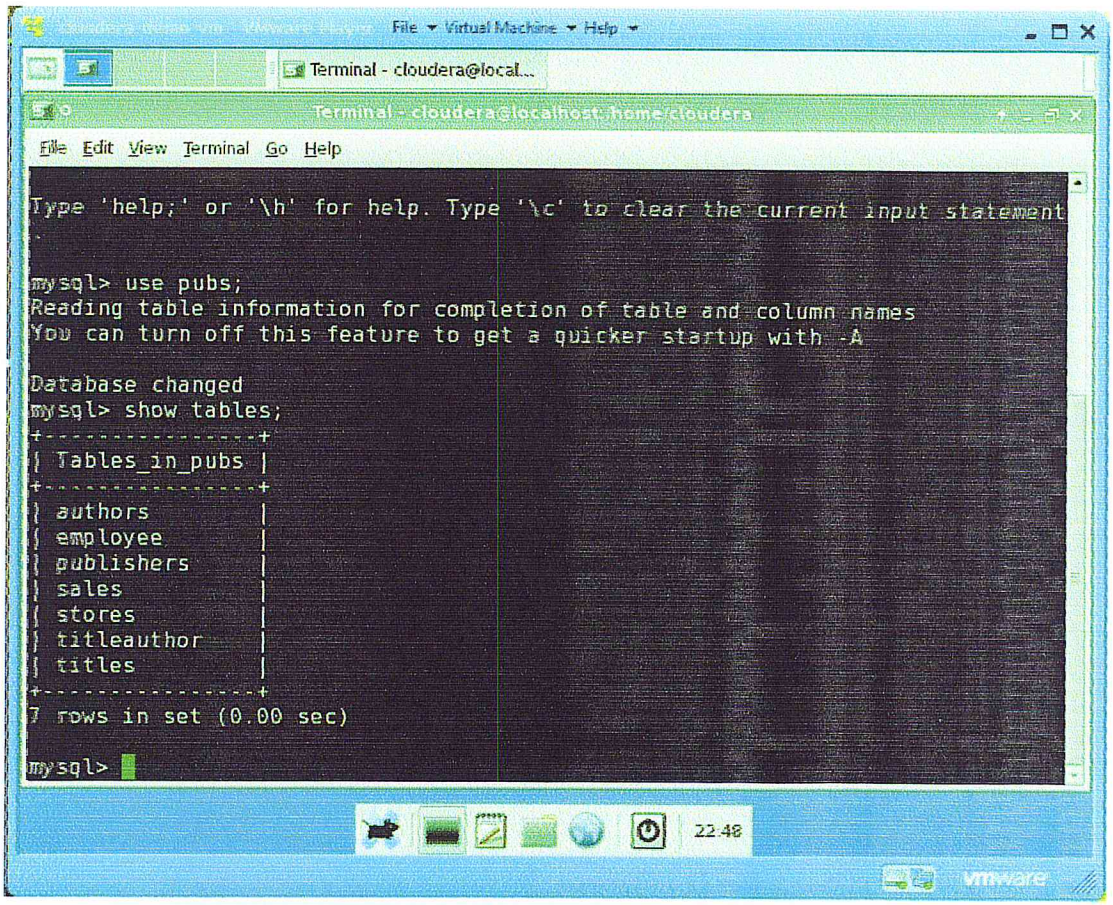

Figure 22 : La base de données pubs

```
mysql> select* from pubs.authors;
+--------+-----------+------------+--------+
| au_id  | au_lname  | au_fname   | state  |
+--------+-----------+------------+--------+
|   213  | Green     | Marjorie   | CA     |
|   238  | Carson    | Cheryl     | CA     |
|   274  | Straight  | Dean       | CA     |
|   409  | Bennet    | Abraham    | CA     |
|   722  | Defrance  | Michel     | IN     |
|   724  | Stringer  | Dirk       | CA     |
|   807  | Panteley  | Sylvia     | MD     |
|   893  | McBadden  | Heather    | CA     |
|   899  | Ringer    | Anne       | UT     |
|   998  | Ringer    | Albert     | UT     |
+--------+-----------+------------+--------+
10 rows in set (0.00 sec)
```

Figure 23 : La table authors (auteur)

```
mysql> select* from pubs.publishers;
+--------+-------------------------+-------+---------+
| pub_id | pub_name                | state | country |
+--------+-------------------------+-------+---------+
|    736 | New Moon Books          | MA    | USA     |
|    877 | Binnet & Hardley        | DC    | USA     |
|   1389 | Algodata Infosystems    | CA    | USA     |
|   1622 | Five Lakes Publishing   | TX    | USA     |
|   9952 | Scootney Books          | NY    | USA     |
+--------+-------------------------+-------+---------+
5 rows in set (0.00 sec)
```

Figure 24 : La table publishers (éditeur)

```
+-------+-----------+-------+
| title_id | title                                          | type
| price | pubdate    | pub_id |
+----------+------------------------------------------------+-------------
+-------+-----------+-------+
| BU1032   | The Busy executive database guide              | business
| 19.99 | 1991-06-12 |   1389 |
| BU1111   | Cooking with computers:Surreptious balance sheets | business
| 11.95 | 1991-06-12 |   1389 |
| MC2222   | Silicon valley gastronomic treats              | mod_cook
| 19.99 | 1991-06-12 |    877 |
| PC1035   | But is it user friendly                        | popular_comp
| 22.95 | 1991-06-30 |   1389 |
| PC8888   | Secret of silicon valley                       | popular_comp
|   20 | 1994-06-12 |   1389 |
| PS1372   | computer phobic and non phobic individuals     | psychology
| 21.59 | 1991-10-21 |    877 |
| PS2091   | Is anger the enemy?                            | psychology
| 10.95 | 1991-06-30 |    736 |
| PS3333   | Prolonged data deprivation:Four case studies   | psychology
| 19.99 | 1991-06-12 |    736 |
| PS7777   | Emotional security:A new algorithm             | psychology
|  7.99 | 1991-06-12 |    736 |
+----------+------------------------------------------------+-------------
```

Figure 25 : La table titles (ouvrages)

```
mysql> select * from pubs.titleauthor;
+-------+----------+
| au_id | title_id |
+-------+----------+
|   213 | BU1032   |
|   409 | BU1032   |
|   724 | BU1111   |
|   238 | PC1035   |
|   998 | PC8888   |
|   899 | PS2091   |
|   998 | PS2091   |
|   213 | PS3333   |
|   213 | PS7777   |
+-------+----------+
9 rows in set (0.00 sec)
```

Figure 26 : La table titleauthor

```
mysql> select* from pubs.sales;
+---------+----------+----------+------------+-----+
| stor_id | ord_num  | title_id | ord_date   | qty |
+---------+----------+----------+------------+-----+
|    6380 | 6871     | BU1032   | 1994-09-14 |   5 |
|    6380 | 722a     | PS2091   | 1994-09-13 |   3 |
|    7066 | QA7442   | PS2091   | 1994-09-13 |  75 |
|    7067 | D4482    | PS2091   | 1994-09-14 |  10 |
|    7131 | N914008  | PS2091   | 1994-09-14 |  20 |
|    8042 | 423LL930 | BU1032   | 1994-09-14 |  10 |
|    8042 | p723     | BU1111   | 1993-03-11 |  25 |
+---------+----------+----------+------------+-----+
7 rows in set (0.00 sec)
```

Figure 27 : La table sales (ventes)

```
mysql> select* from pubs.stores;
+---------+-----------------------------+-------+
| stor_id | stor_name                   | state |
+---------+-----------------------------+-------+
|    6380 | Eric the read books         | WA    |
|    7066 | Barnum                      | CA    |
|    7067 | News & Brews                | CA    |
|    7131 | Quality laundry and Books   | WA    |
|    7896 | Fricative Bookshop          | CA    |
|    8042 | Bookbeat                    | OR    |
+---------+-----------------------------+-------+
6 rows in set (0.00 sec)
```

Figure 28 : La table stores (magasins)

```
mysql> select * from pubs.employee;
+--------+-----------+---------+--------+
| emp_id | fname     | lname   | pub_id |
+--------+-----------+---------+--------+
| AM     | Ann       | Devon   |   9952 |
| FC     | Francisco | Chang   |   9952 |
| LA     | Laurence  | Lebihan |    736 |
| PT     | Philip    | Cramer  |   9952 |
| PX     | Paul      | Henriot |    877 |
| RB     | NULL      | NULL    |   1622 |
| SK     | Sven      | NULL    |   1389 |
+--------+-----------+---------+--------+
7 rows in set (0.00 sec)
```

Figure 29 : La table employee (employé)

Etape 3: La base de données créée dans l'étape précédente représente les données sources.

Dans cette étape, nous allons extraire, transformer ces données avec MapReduce et les mettre dans le Hadoop Distributed File System HDFS. Pour cela, nous utilisons l'outil Sqoop qui est conçu pour importer et exporter les données entre Hadoop et les systèmes de gestion de base de données relationnelle (SGBDR) comme MySQL. Sqoop utilise MapReduce qui fournit un fonctionnement en parallèle ainsi qu'une tolérance aux pannes.

« Sqoop appelé aussi SQL-to-Hadoop, il fait partie en Mars 2012 des projets Apache Software Foundation (ASF). Cet outil peut orienter les données vers d'autres applications Hadoop comme le système de gestion de base de données non relationnelle HBase et son stockage structuré pour les grandes tables, ou vers le logiciel Hive. » **[Web 6]** comme le montre la figure suivante :

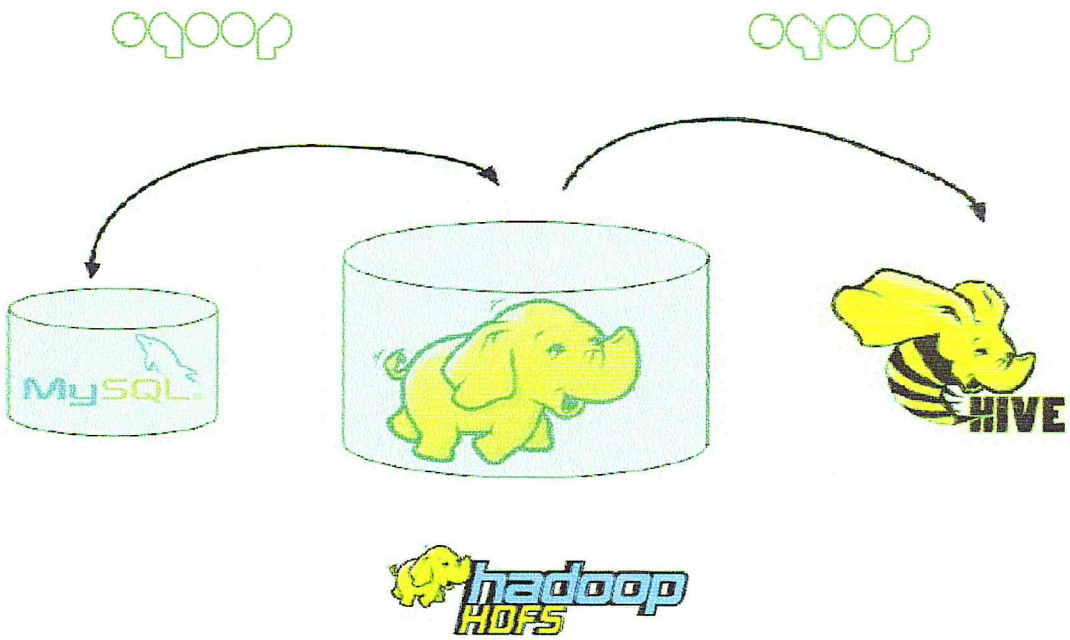

Figure 30 : Sqoop SQL-to-Hadoop

L'entrée dans le processus d'importation est une table d'une base de données. Sqoop lis la table ligne par ligne, la partitionne selon la clé primaire et la met dans HDFS.

La sortie de ce processus d'importation est un ensemble de fichiers contenant une copie de la table importée. La procédure d'importation est effectuée en parallèle selon l'algorithme MapReduce. Pour cette raison, la sortie sera dans plusieurs fichiers qui peuvent être des fichiers texte délimités (par exemple, avec des virgules ou des tabulations séparant chaque champ), ou binaire Avro ou SequenceFiles contenant des données d'enregistrement sérialisées.

Les figures suivantes illustrent l'extraction des données avec Sqoop :

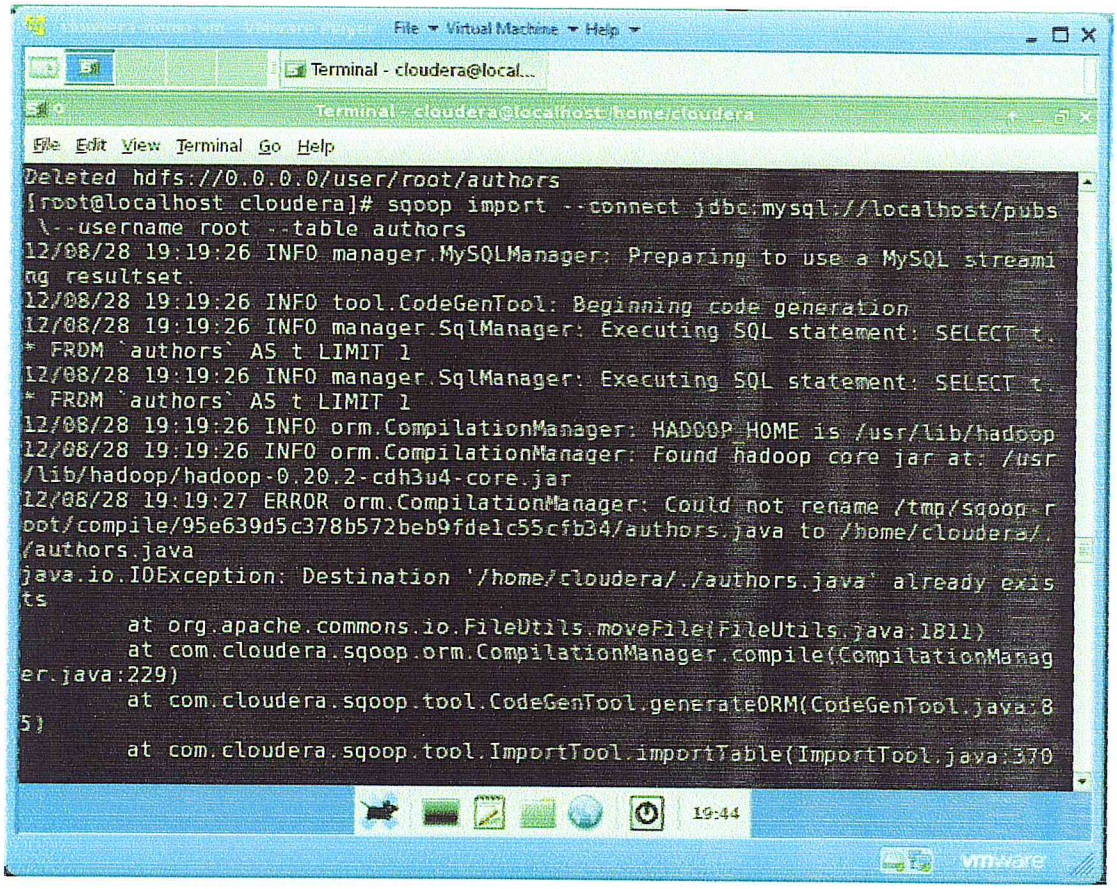

Figure 31 : Importer la table authors avec Sqoop (1)

Comme illustré dans cette figure, la syntaxe d'extraction des données avec Sqoop est :

sqoop import –connect jdbc:mysql://localhost/pubs \ --username root –table authors

sqoop import : signifie que nous allons utilisé Sqoop pour une importation.

–connect jdbc:mysql ://localhost/ : pour se connecter à MySQL.

pubs : le nom de notre base de données sous MySQL.

--username : le username de MySQL, dans ce cas c'est root.

–table authors : le nom de la table à importer de la base de données pubs.

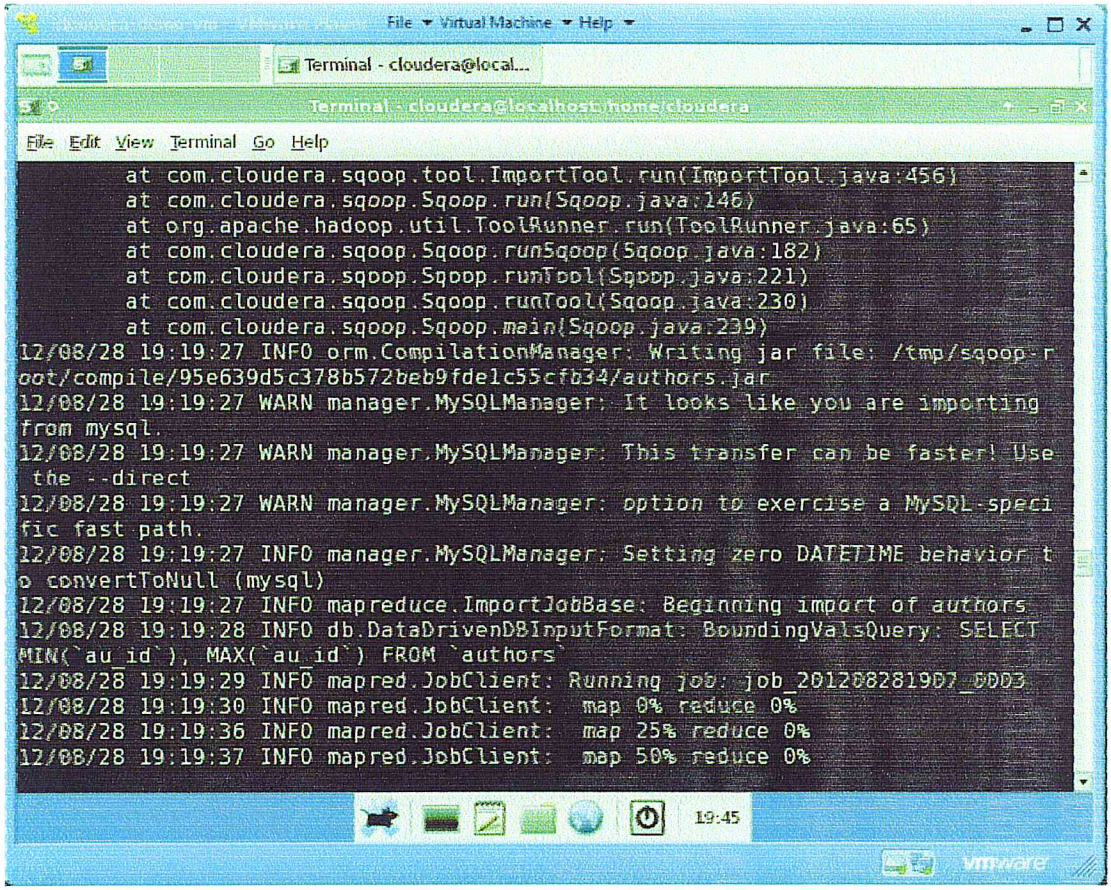

Figure 32 : Importer la table authors avec Sqoop (2)

Cette figure montre le partitionnement de la table authors selon la clé primaire avec l'expression :

SELECT MIN('au_id'), MAX('au_id') FROM 'authors'

Une fois le partitionnement terminé, Sqoop fait appel au mapper et reducer pour extraire la table authors.

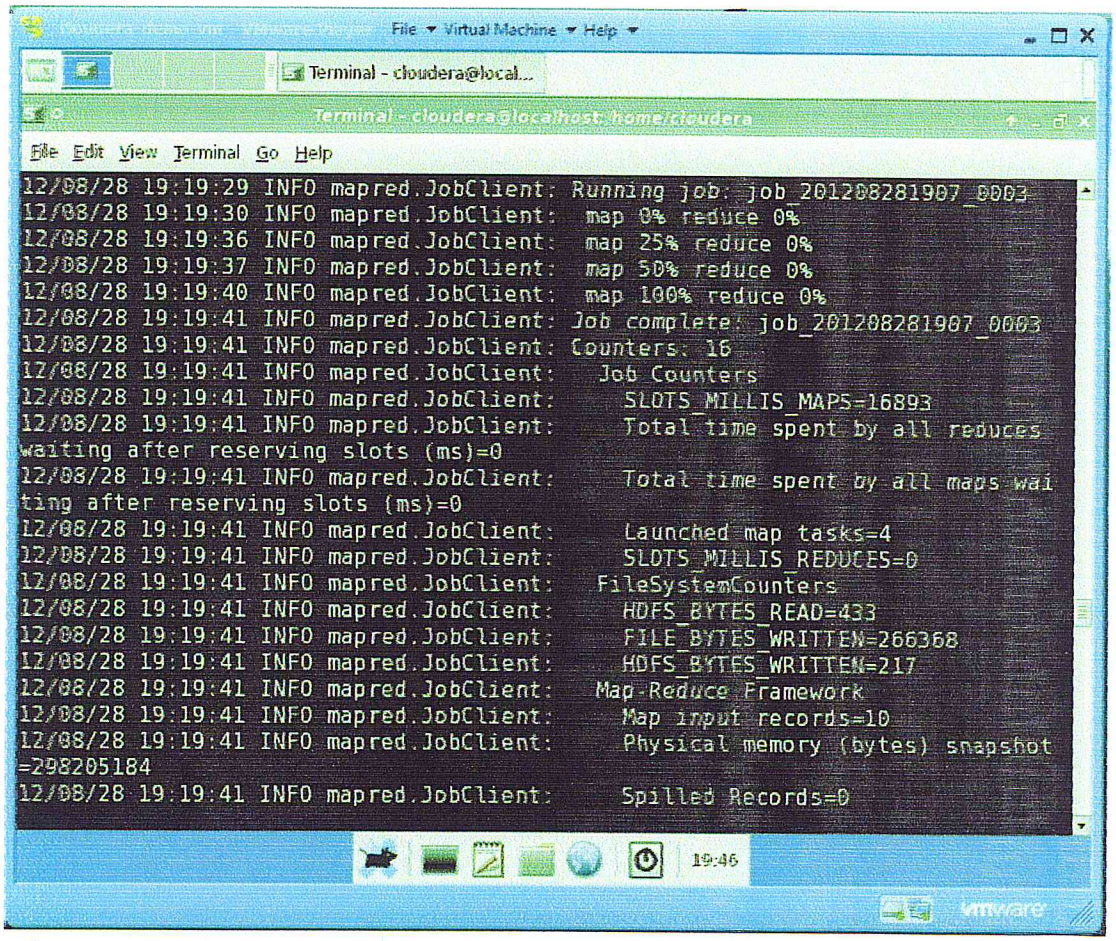

Figure 33 : Importer la table authors avec Sqoop (3)

Cette figure montre le processus d'importation de la table authors selon l'algorithme MapReduce.

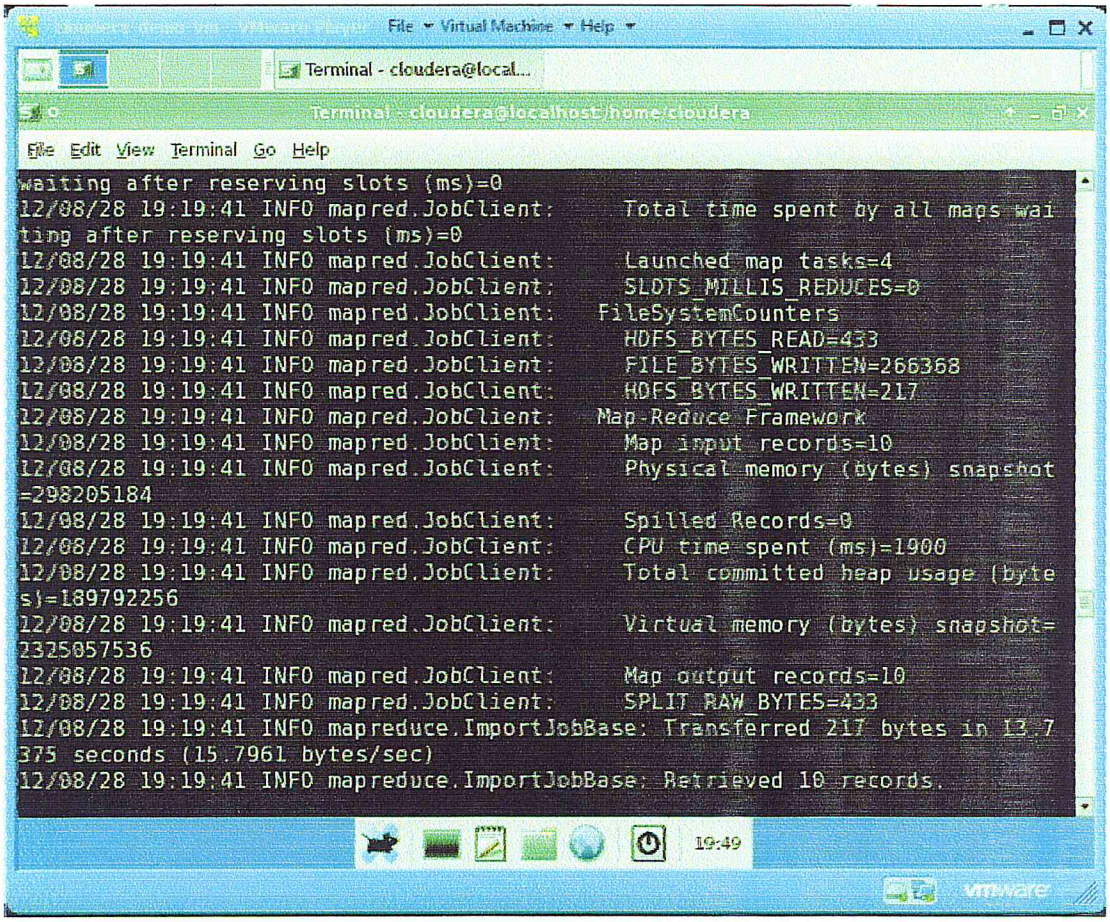

Figure 34 : Importer la table authors avec Sqoop (4)

Pour une extraction en parallèle, nous remarquons dans cette figure que Sqoop a utilisé quatre mapper et zéro reducer. Donc la table a été importée en quatre phases. Comme résultat, nous trouvons quatre fichiers texte dans HDFS. Comme le montre la figure suivante :

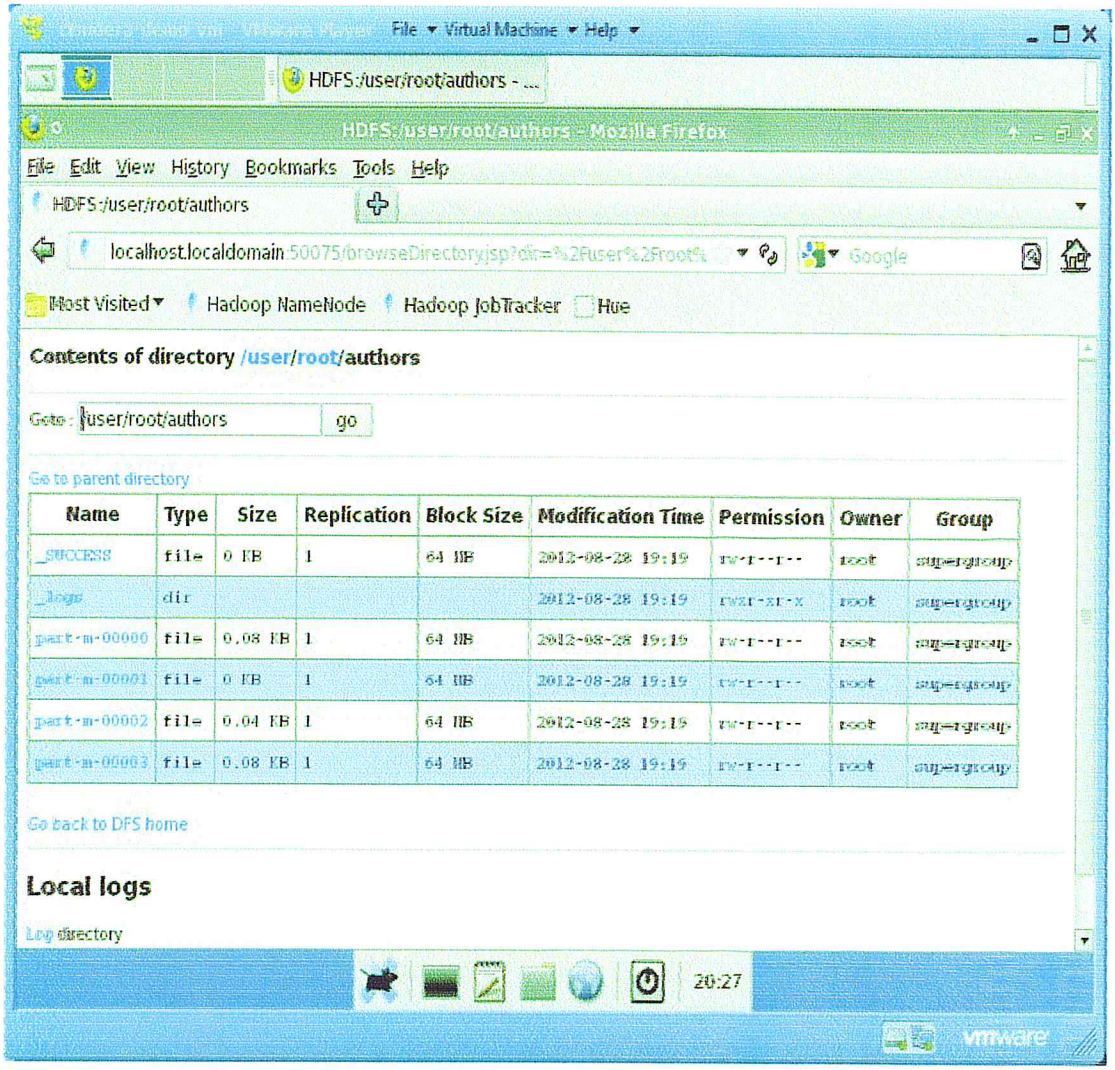

Figure 35 : Résultat d'importation de la table authors avec Sqoop

Chaque fichier contient une partie des enregistrements de la table authors, comme l'illustrent les figures suivantes :

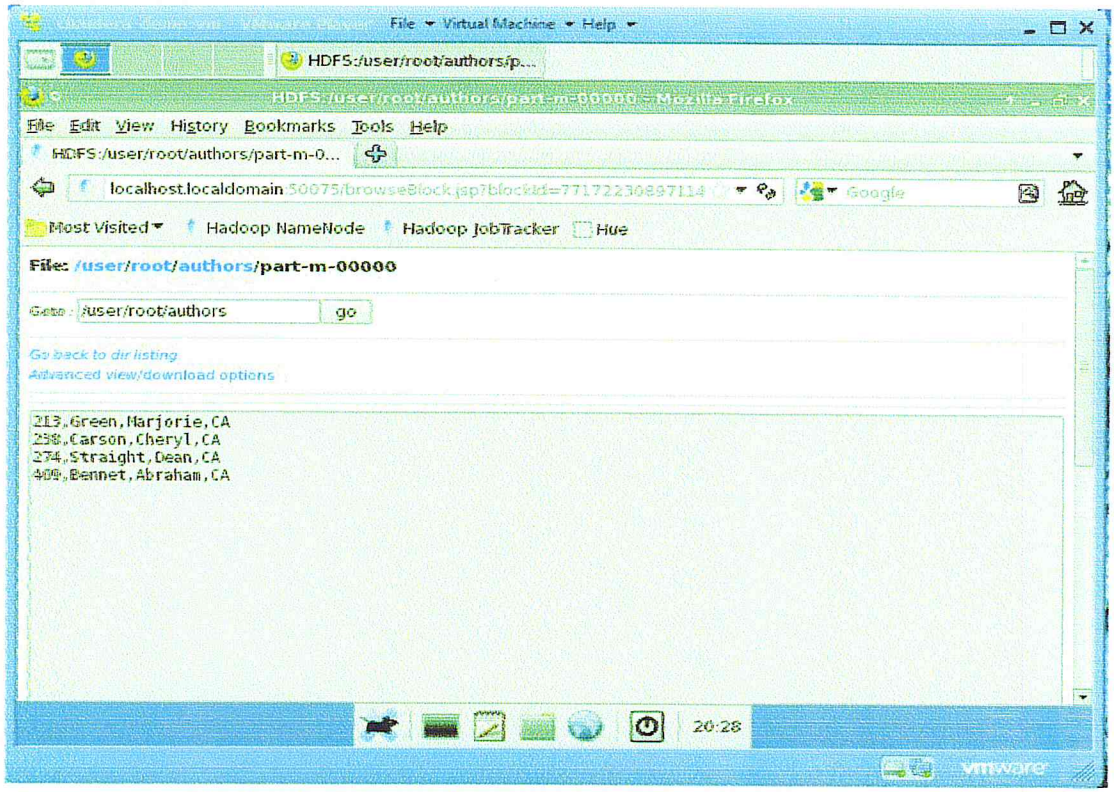

Figure 36 : La table authors dans HDFS fichier 1

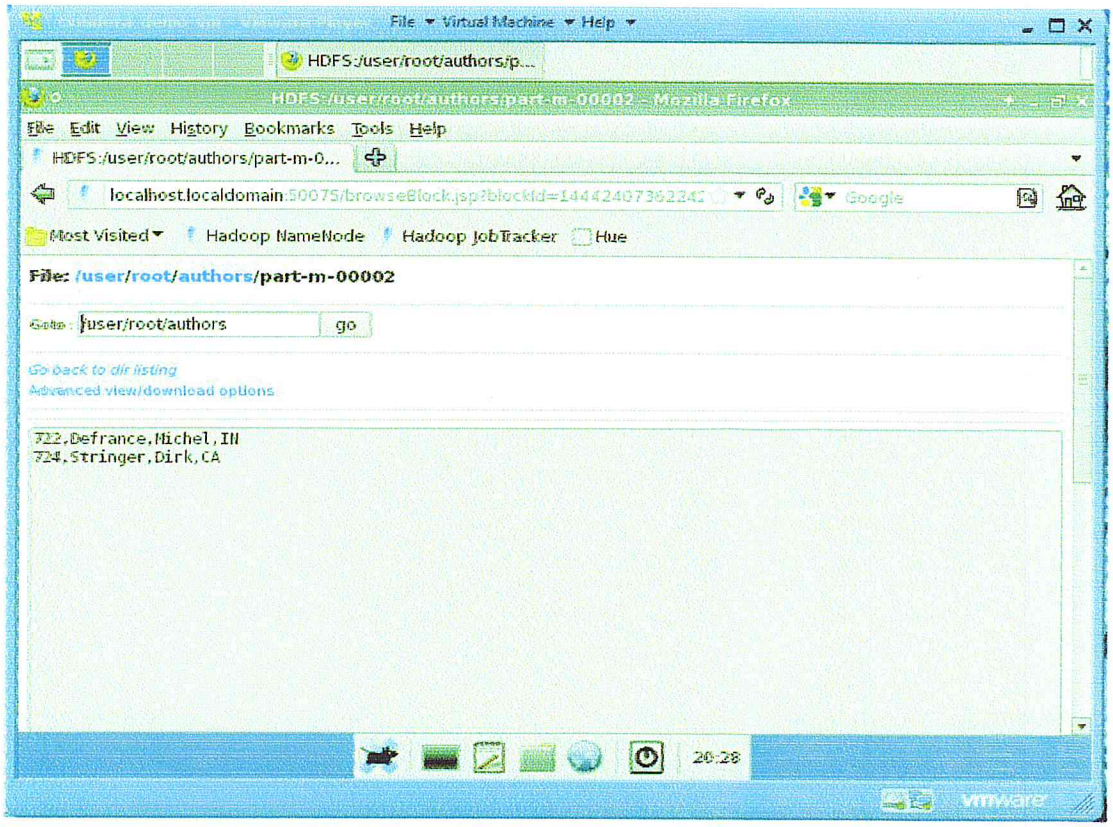

Figure 37 : La table authors dans HDFS fichier 2

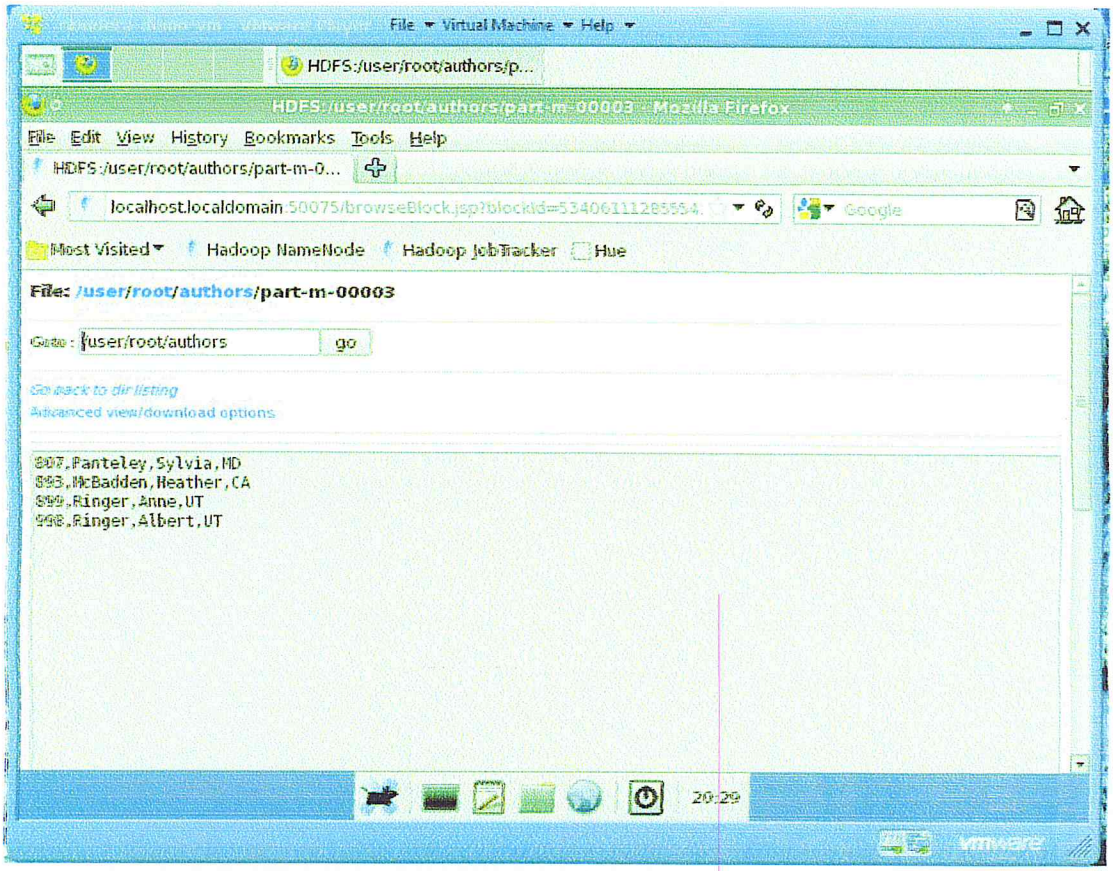

Figure 38 : La table authors dans HDFS fichier 3

Avec l'outil Sqoop nous pouvons fixer le nombre de mapper utilisé lors de l'extraction d'une table en ajoutant à la fin de la syntaxe d'importation :
-m nbr_mapper , par exemple : -m 1, -m 2, etc.

-m 1 spécifie une tâche map unique, donc la requête est exécutée une seule fois et la table est importée en série. Comme suit :

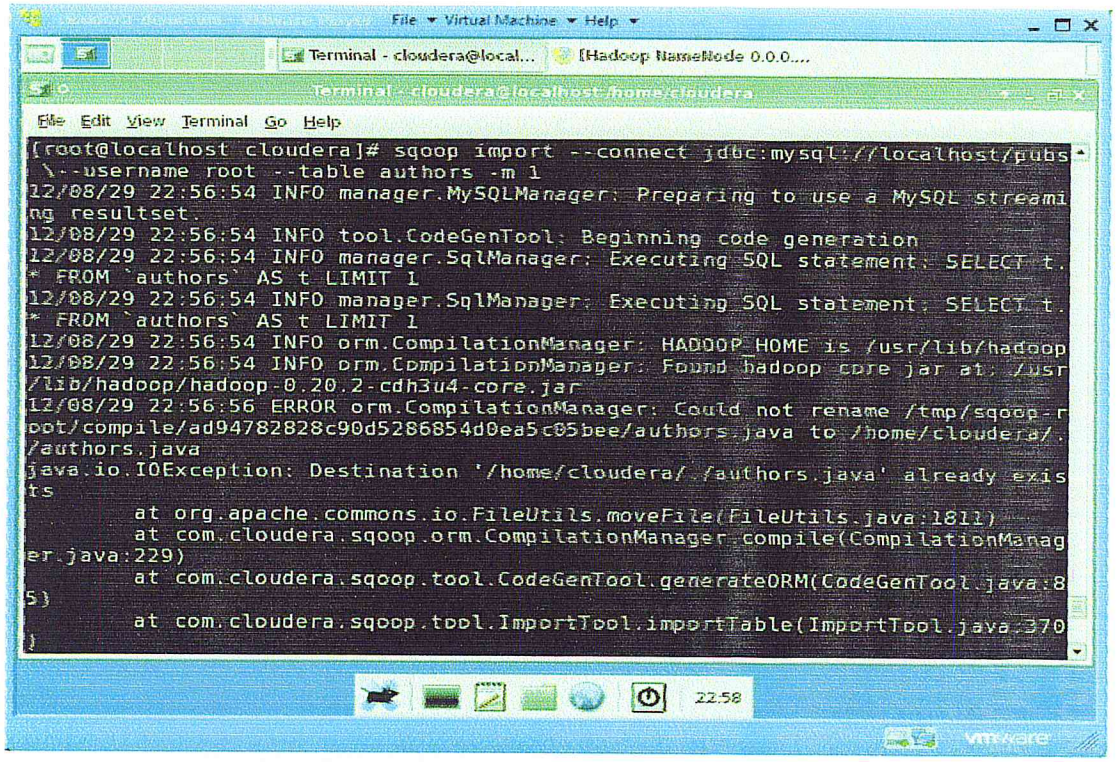

Figure 39 : Importation de la table authors avec Sqoop (map=1) (1)

Cette figure illustre l'importation de la table authors en fixant le nombre de mapper à un (1).

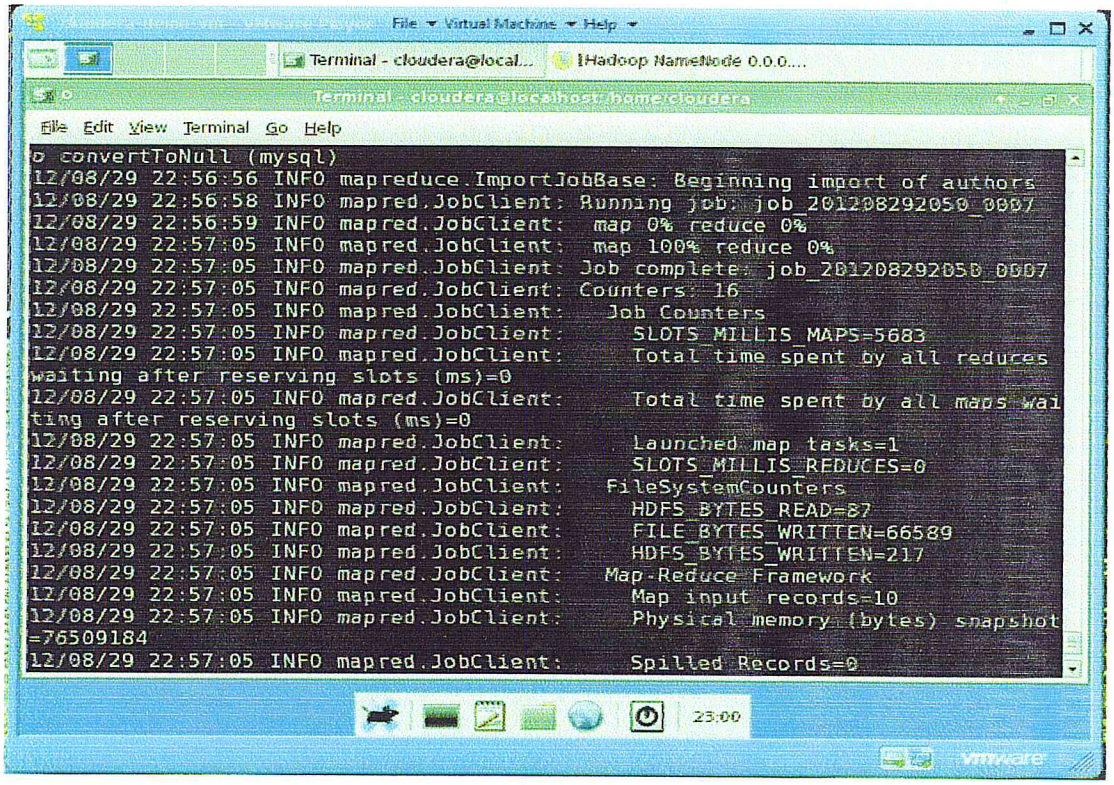

Figure 40 : Importation de la table authors avec Sqoop (map=1) (2)

Dans cette figure, nous remarquons que Sqoop a utilisé une seule tâche map pour importer la table authors.

Le résultat de cette importation avec une seule tâche map est un seul fichier texte de la table authors dans HDFS comme l'illustre la figure suivante :

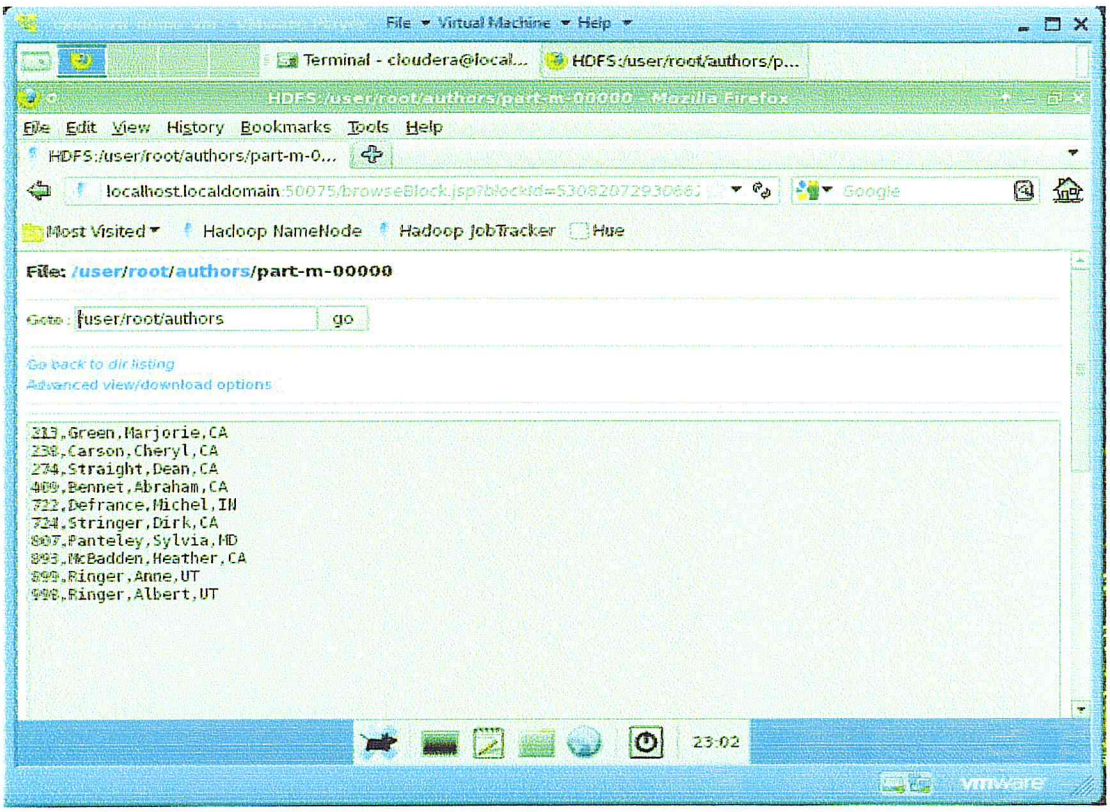

Figure 41 : Résultat d'importation de la table authors avec Sqoop (map=1)

Avec l'outil Sqoop nous pouvons aussi importer des tables de MySQL à Hive et c'est ce qui nous intéresse. Pour cela nous ajoutant –hive-import à la fin de l'expression sqoop import.

Comme exemple, nous important la table 'a' de MySQL à Hive en spécifiant le nombre de mapper égal à deux. (la table 'a' est la même que authors).

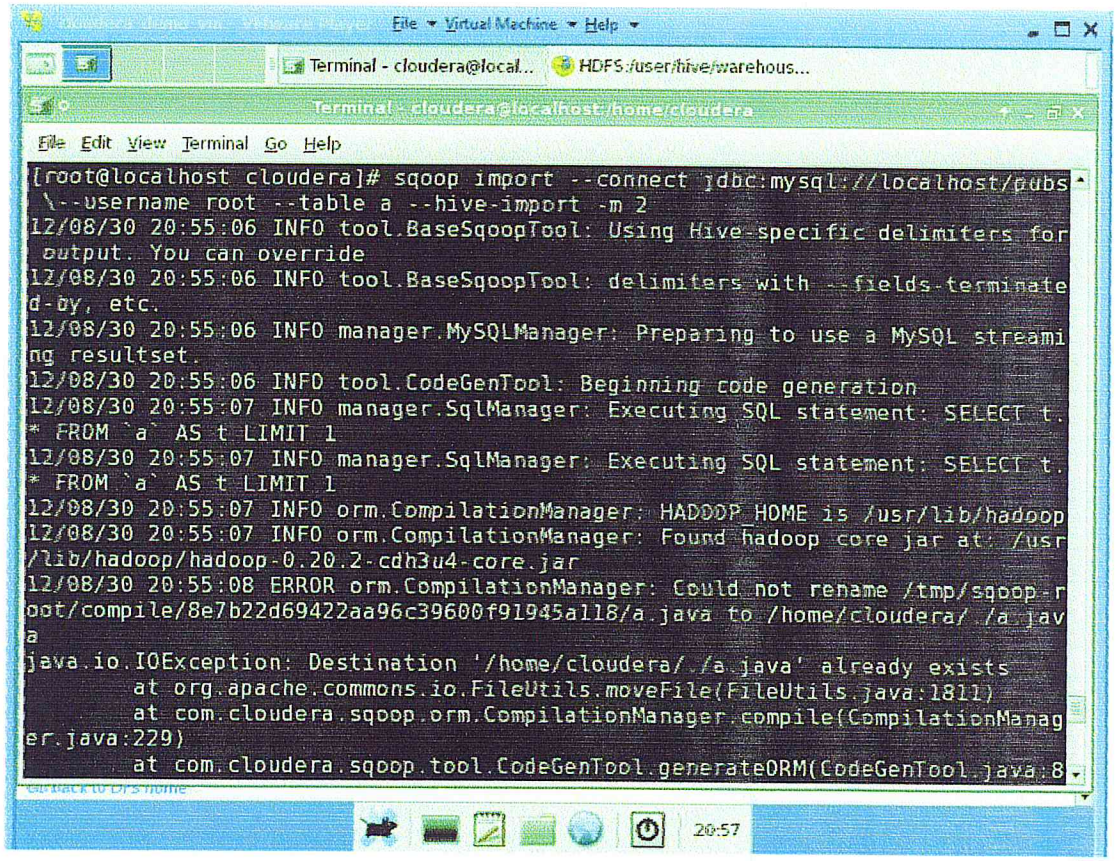

Figure 42 : Importation de la table a avec Sqoop vers Hive (map=2) (1)

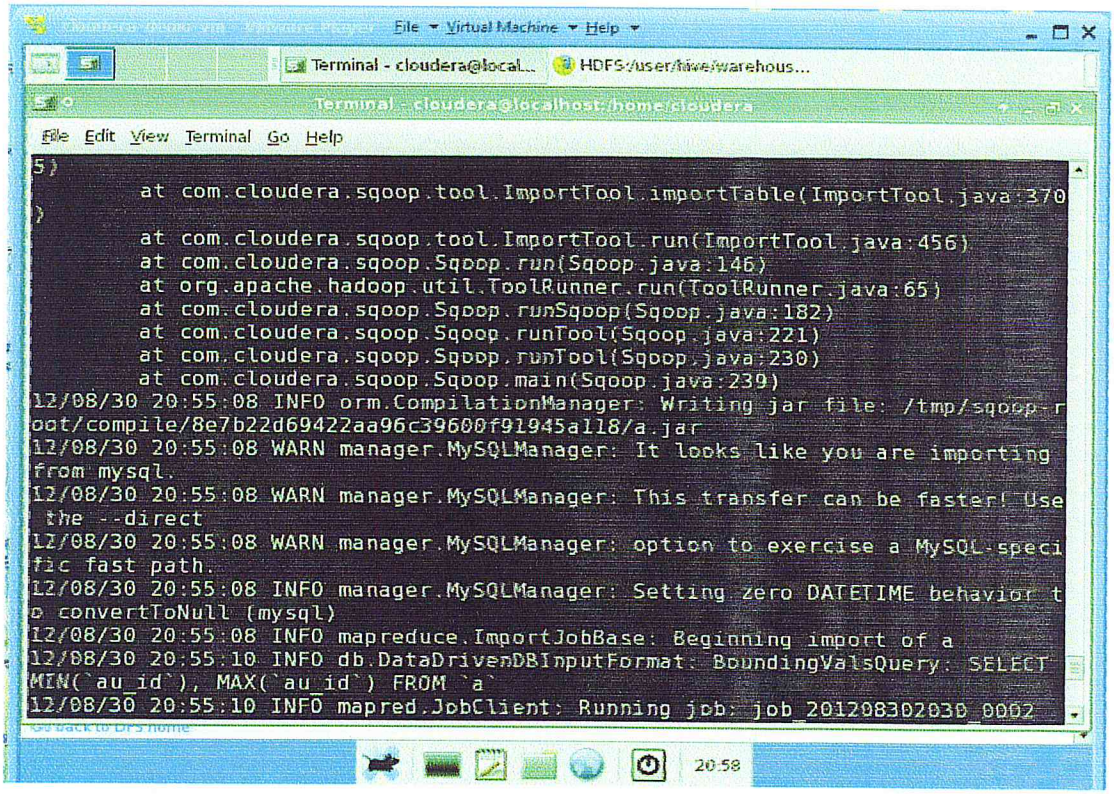

Figure 43 : Importation de la table a avec Sqoop vers Hive (map=2) (2)

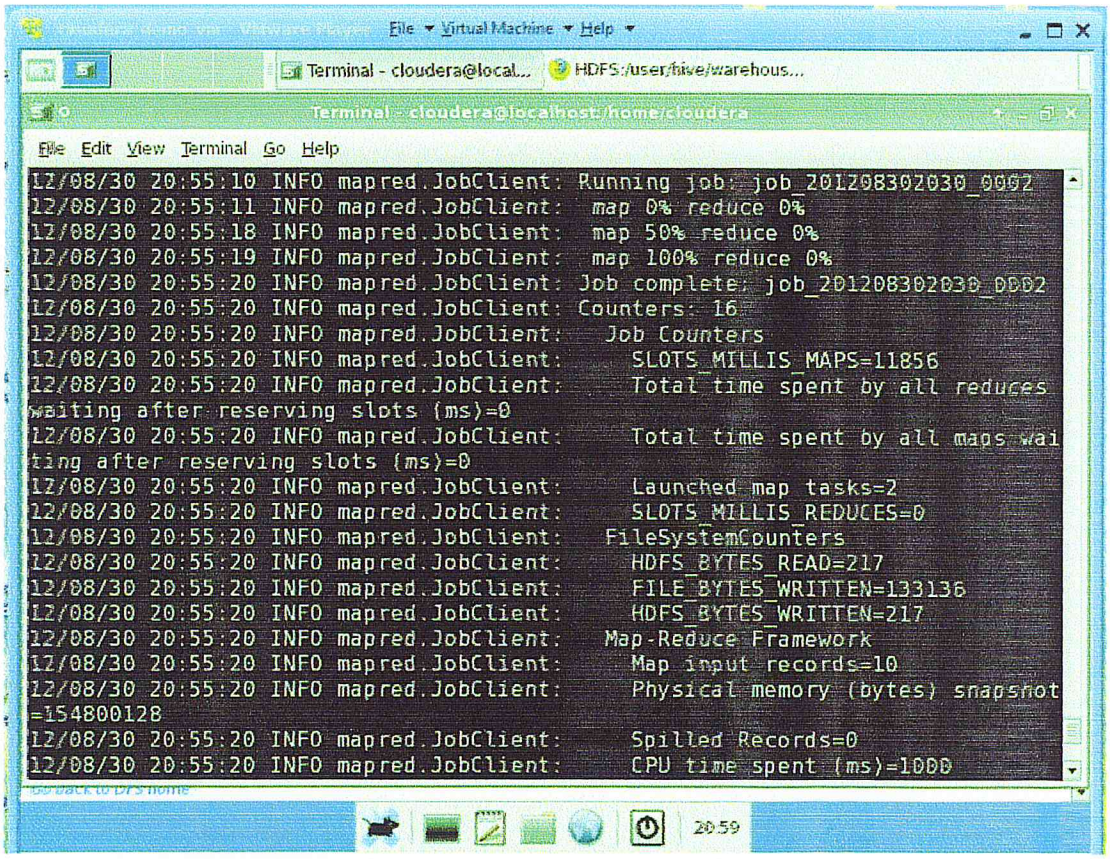

Figure 44 : Importation de la table a avec Sqoop vers Hive (map=2) (3)

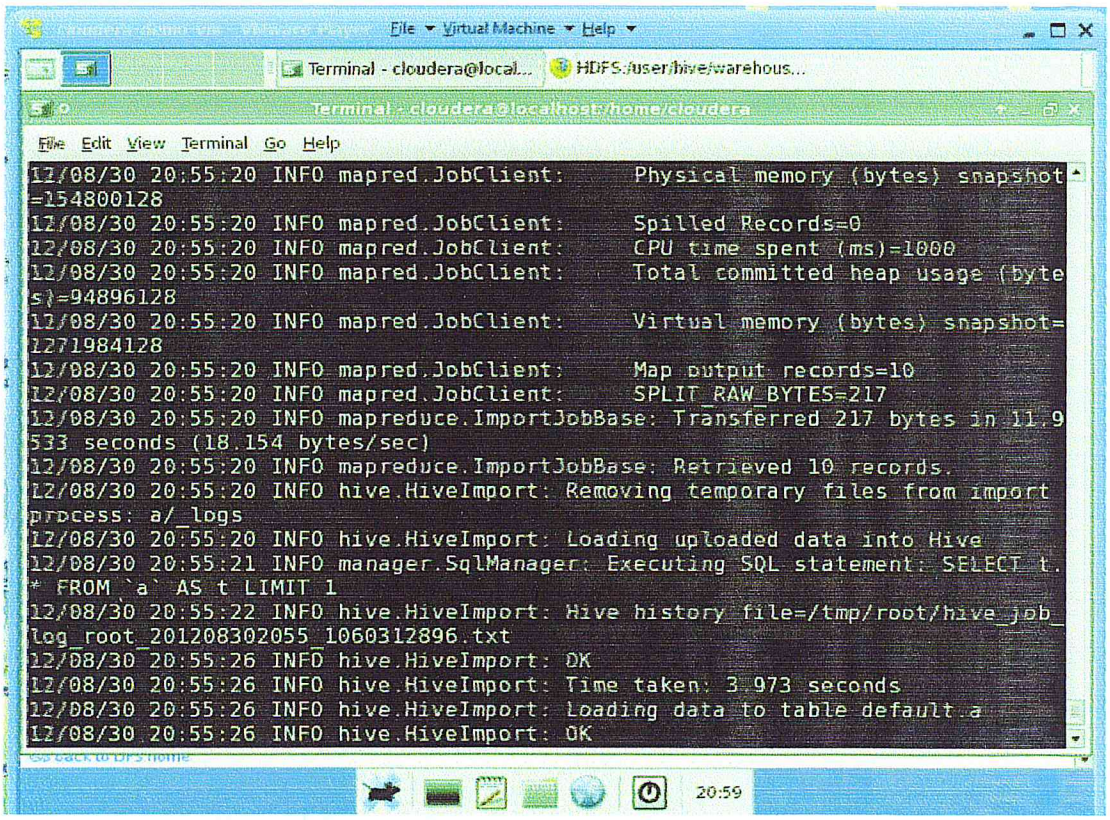

Figure 45 : Importation de la table a avec Sqoop vers Hive (map=2) (4)

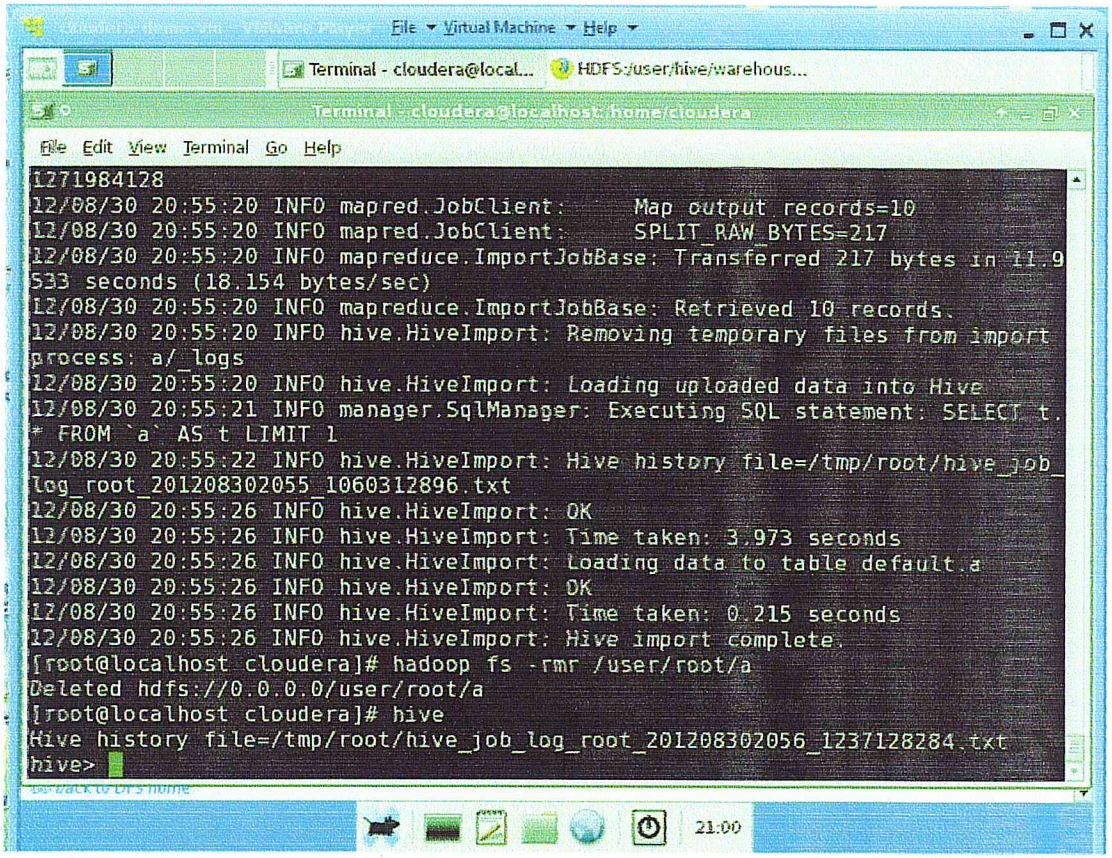

Figure 46 : Importation de la table a avec Sqoop vers Hive (map=2) (5)

Le résultat d'importation de la table 'a' est deux fichiers qui contiennent les enregistrements de la table 'a'. Ils sont stockés dans le répertoire /user/hive/warehouse qui est le répertoire par défaut de Hive dans HDFS. Ces deux fichiers sont illustrés par la figure suivante :

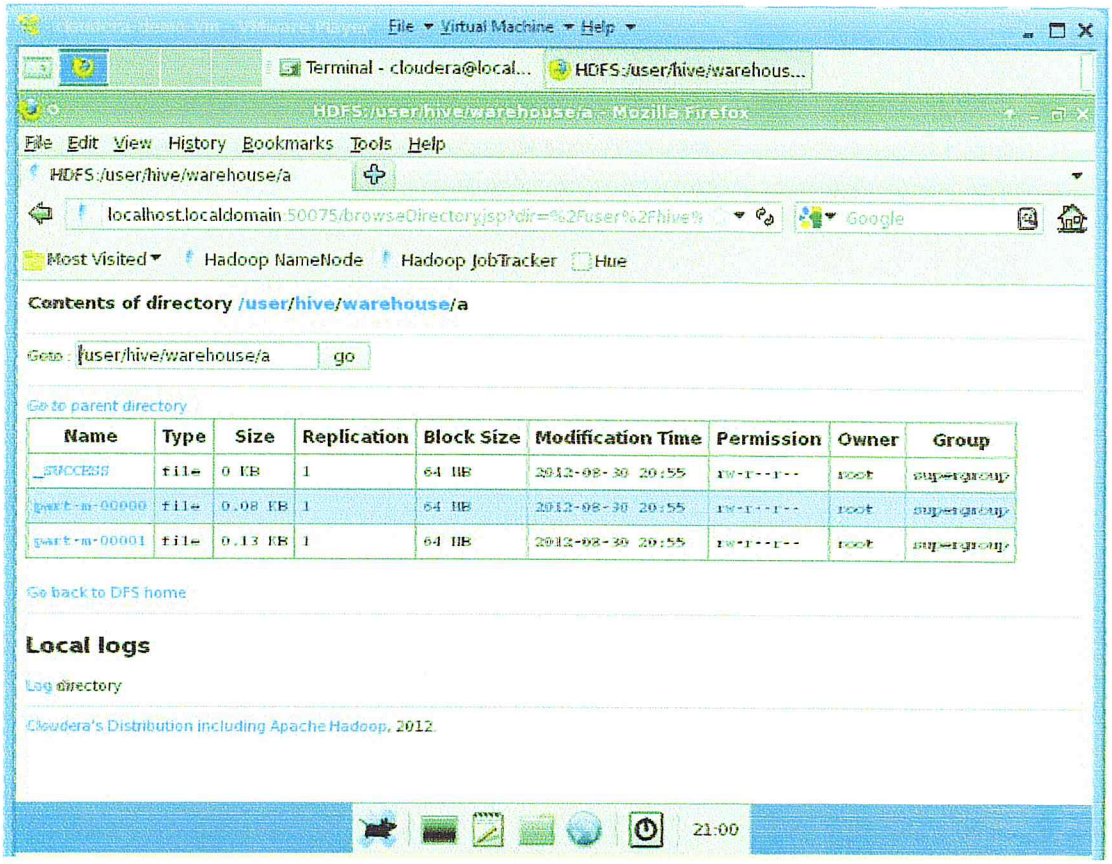

Figure 47 : Résultat d'importation de la table 'a' avec Sqoop vers Hive (map=2)

Et ainsi de suite nous importons toutes les tables de notre base de données pubs vers Hive. Sauf la table employee qui contient des enregistrements avec des valeurs null. Pour transformer cette table afin d'importer que les données significatives, nous pouvons spécifier avec Sqoop une instruction SQL avec l'argument --query. Comme le montre la figure suivante :

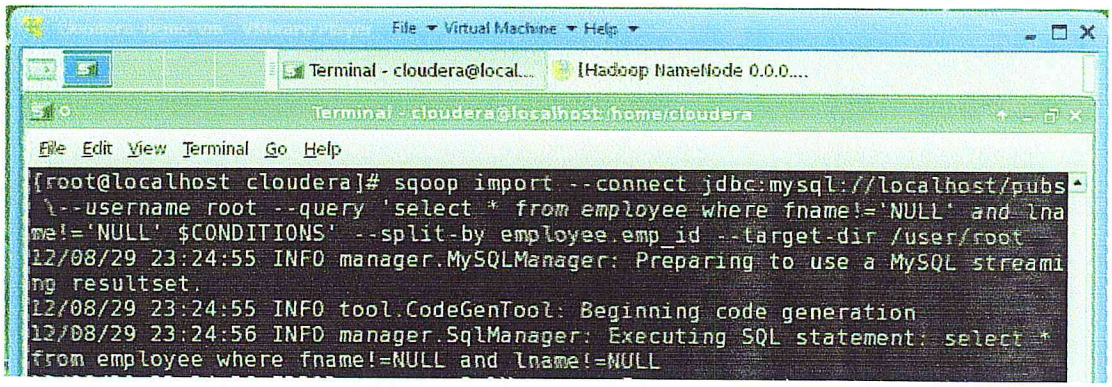

Figure 48 : Importation et transformation de la table employee avec Sqoop

Le résultat de cette requête est le fichier suivant :

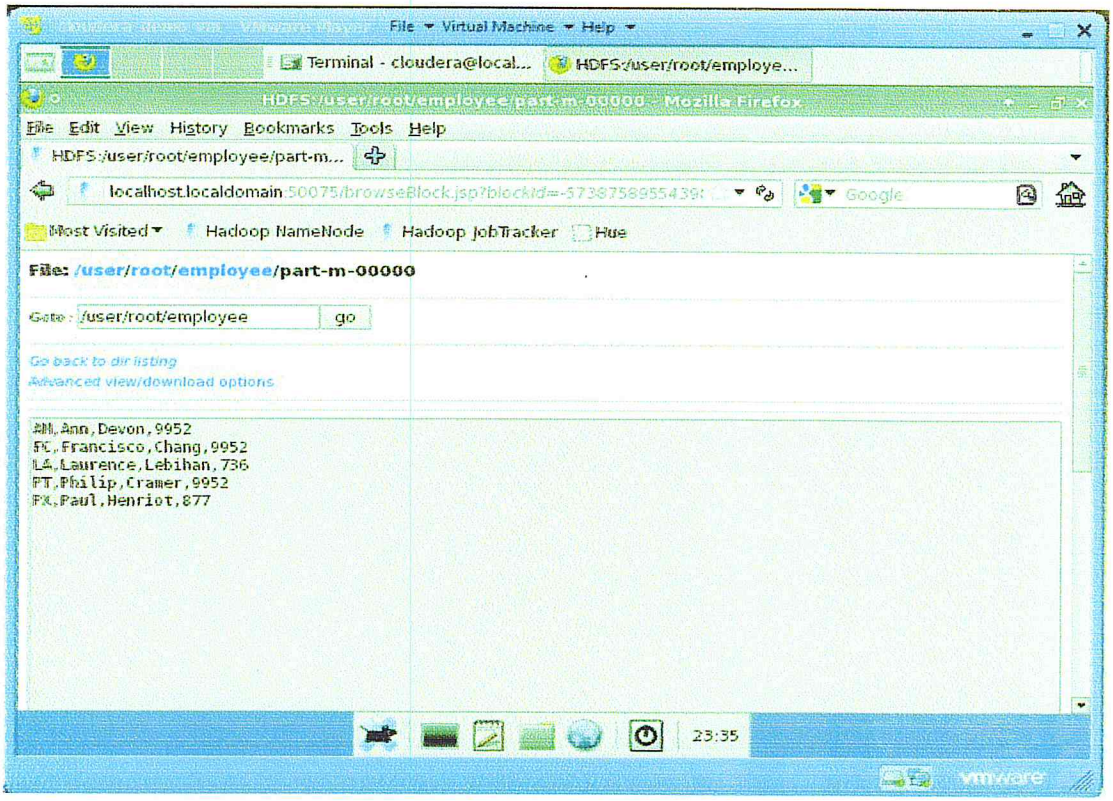

Figure 49 : Résultat d'importation et transformation de la table employee avec Sqoop

A la fin de cette étape, les tables de notre base de données sont dans Hive (sous forme de fichiers textes) qui représente notre ODS (Operational Data Store) comme le montre la figure suivante :

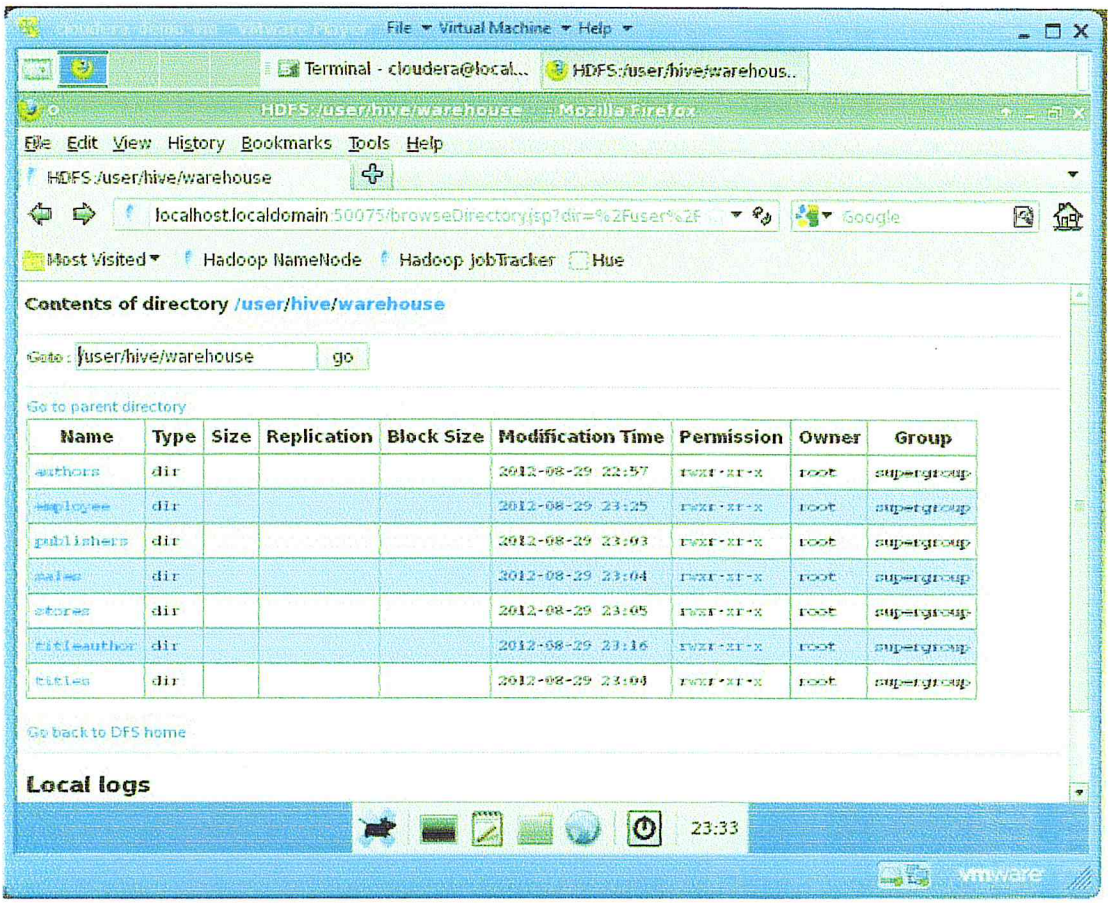

Figure 50 : Importation des tables de la base de données pubs dans HDFS

Etape 4 : Une fois nos données sources dans Hive, nous passons à cette étape qui consiste en la création des tables de faits et dimensions en utilisant le langage HiveQL de Hive. Les requêtes HiveQL sont proches de celles de SQL sauf qu'elles s'exécutent en MapReduce.

Une table créée sous Hive est stockée dans le répertoire /user/hive/warehouse.

Pour pouvoir remplir les tables de dimensions à partir de nos tables sources, il faut que ces deux tables soient dans le même répertoire.

Notons que nous pouvons remplir une table à partir d'une autre table importée par Sqoop et stockée en plusieurs fichiers dans HDFS (Hive) car il existe un lien entre ces fichiers et ils sont tous reliés à la table importée.

Comme exemple de schéma en étoile, nous calculons le chiffre d'affaire des ventes des ouvrages selon publishers, titles et sales. Les figures suivantes montrent la création et le remplissage des tables nécessaires pour la construction de notre schéma.

```
hive> create external table td_publishers (pub_id int, pub_name string, state
 string, country string) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES
TERMINATED BY '\n' STORED AS TEXTFILE;
OK
Time taken: 0.158 seconds
```

Figure 51 : Création de la table de dimension td_publishers

```
hive> create table td_titles (title_id string, title string, type string, pri
ce float, pubdatestring,pub_id int) ROW FORMAT DELIMITED FIELDS TERMINATED BY
 ',' LINES TERMINATED BY '\n' STORED AS TEXTFILE;
OK
Time taken: 0.142 seconds
```

Figure 52 : Création de la table de dimension td_titles

```
hive> create external table td_sales (stor_num int, ord_num string, title_id
string, ord_date string, qty int) ROW FORMAT DELIMITED FIELDS TERMINATED BY '
,' LINES TERMINATED BY '\n' STORED AS TEXTFILE;
OK
Time taken: 0.098 seconds
```

Figure 53 : Création de la table de dimension td_sales

```
hive> create external table tf_ca (pub_id int,  title_id string,
 stor_id int, ca float) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES T
ERMINATED BY '\n' STORED AS TEXTFILE;
OK
Time taken: 0.142 seconds
```

Figure 54 : Création de la table de fait tf_ca

```
hive> insert overwrite table td_publishers select * from publishers;
Total MapReduce jobs = 2
Launching Job 1 out of 2
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_201208291151_0015, Tracking URL = http://0.0.0.0:50030/job
details.jsp?jobid=job_201208291151_0015
Kill Command = /usr/lib/hadoop/bin/hadoop job  -Dmapred.job.tracker=0.0.0.0:8
021 -kill job_201208291151_0015
2012-08-29 14:30:29,633 Stage-1 map = 0%,  reduce = 0%
2012-08-29 14:30:31,655 Stage-1 map = 100%,  reduce = 0%
2012-08-29 14:30:53,971 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_201208291151_0015
Ended Job = -222883, job is filtered out (removed at runtime).
Moving data to: hdfs://0.0.0.0/tmp/hive-root/hive_2012-08-29_14-30-24_008_700
5299577831776332/-ext-10000
Loading data to table default.td_publishers
Deleted hdfs://0.0.0.0/user/hive/warehouse/td_publishers
Table default.td_publishers stats: [num_partitions: 0, num_files: 1, num_rows
: 0, total_size: 148]
5 Rows loaded to td_publishers
OK
Time taken: 30.319 seconds
```

Figure 55 : Remplissage de la table td_publishers

Cette figure montre le remplissage de la table de dimension td_publishers en MapReduce.

```
hive> insert overwrite table td_titles select * from titles;
Total MapReduce jobs = 2
Launching Job 1 out of 2
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_201208291151_0016, Tracking URL = http://0.0.0.0:50030/job
details.jsp?jobid=job_201208291151_0016
Kill Command = /usr/lib/hadoop/bin/hadoop job  -Dmapred.job.tracker=0.0.0.0:8
021 -kill job_201208291151_0016
2012-08-29 14:33:10,760 Stage-1 map = 0%,  reduce = 0%
2012-08-29 14:33:12,787 Stage-1 map = 100%,  reduce = 0%
2012-08-29 14:33:14,808 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_201208291151_0016
Ended Job = -1223021726, job is filtered out (removed at runtime).
Moving data to: hdfs://0.0.0.0/tmp/hive-root/hive_2012-08-29_14-33-02_087_193
5931170548170290/-ext-10000
Loading data to table default.td_titles
Deleted hdfs://0.0.0.0/user/hive/warehouse/td_titles
Table default.td_titles stats: [num_partitions: 0, num_files: 1, num_rows: 0,
 total_size: 661]
9 Rows loaded to td_titles
OK
Time taken: 13.085 seconds
```

Figure 56 : Remplissage de la table td_titles

```
hive> insert overwrite table td_sales select * from sales;
Total MapReduce jobs = 2
Launching Job 1 out of 2
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_201208291151_0018, Tracking URL = http://0.0.0.0:50030/job
details.jsp?jobid=job_201208291151_0018
Kill Command = /usr/lib/hadoop/bin/hadoop job  -Dmapred.job.tracker=0.0.0.0:8
021 -kill job_201208291151_0018
2012-08-29 15:02:22,058 Stage-1 map = 0%,  reduce = 0%
2012-08-29 15:02:26,096 Stage-1 map = 100%,  reduce = 0%
2012-08-29 15:02:30,142 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_201208291151_0018
Ended Job = -2034515821, job is filtered out (removed at runtime).
Moving data to: hdfs://0.0.0.0/tmp/hive-root/hive_2012-08-29_15-02-16_476_174
7142513553313373/-ext-10000
Loading data to table default.td_sales
Deleted hdfs://0.0.0.0/user/hive/warehouse/td_sales
Table default.td_sales stats: [num_partitions: 0, num_files: 1, num_rows: 0,
total_size: 225]
7 Rows loaded to td_sales
OK
Time taken: 13.899 seconds
```

Figure 57 : Remplissage de la table td_sales

```
hive> insert overwrite table tf_ca
    > select p.pub_id,t.title_id,s.stor_id,sum(s.qty*t.price)
    > from publishers p
    > join titles t on (p.pub_id=t.pub_id)
    > join sales s on (t.title_id=s.title_id)
    > GROUP BY p.pub_id,t.title_id,s.stor_id
    > ;
Total MapReduce jobs = 3
Launching Job 1 out of 3
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_201208291151_0026, Tracking URL = http://0.0.0.0:50030/job
details.jsp?jobid=job_201208291151_0026
Kill Command = /usr/lib/hadoop/bin/hadoop job  -Dmapred.job.tracker=0.0.0.0:8
021 -kill job_201208291151_0026
2012-08-29 19:53:40,079 Stage-1 map = 0%,  reduce = 0%
2012-08-29 19:53:44,110 Stage-1 map = 100%,  reduce = 0%
2012-08-29 19:53:51,196 Stage-1 map = 100%,  reduce = 33%
2012-08-29 19:53:52,202 Stage-1 map = 100%,  reduce = 100%
```

Figure 58 : Remplissage de la table tf_ca (1)

```
Ended Job = job_201208291151_0026
Launching Job 2 out of 3
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_201208291151_0027, Tracking URL = http://0.0.0.0:50030/job
details.jsp?jobid=job_201208291151_0027
Kill Command = /usr/lib/hadoop/bin/hadoop job  -Dmapred.job.tracker=0.0.0.0:8
021 -kill job_201208291151_0027
2012-08-29 19:54:00,435 Stage-2 map = 0%,  reduce = 0%
2012-08-29 19:54:04,464 Stage-2 map = 100%,  reduce = 0%
2012-08-29 19:54:11,540 Stage-2 map = 100%,  reduce = 33%
2012-08-29 19:54:12,546 Stage-2 map = 100%,  reduce = 100%
Ended Job = job_201208291151_0027
Launching Job 3 out of 3
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
```

Figure 59 : Remplissage de la table tf_ca (2)

```
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_201208291151_0028, Tracking URL = http://0.0.0.0:50030/job
details.jsp?jobid=job_201208291151_0028
Kill Command = /usr/lib/hadoop/bin/hadoop job  -Dmapred.job.tracker=0.0.0.0:8
021 -kill job_201208291151_0028
2012-08-29 19:54:18,237 Stage-3 map = 0%,  reduce = 0%
2012-08-29 19:54:20,250 Stage-3 map = 100%,  reduce = 0%
2012-08-29 19:54:27,332 Stage-3 map = 100%,  reduce = 33%
2012-08-29 19:54:29,348 Stage-3 map = 100%,  reduce = 100%
Ended Job = job_201208291151_0028
Loading data to table default.tf_ca
Deleted hdfs://0.0.0.0/user/hive/warehouse/tf_ca
Table default.tf_ca stats: [num_partitions: 0, num_files: 1, num_rows: 0, tot
al_size: 159]
7 Rows loaded to tf_ca
OK
Time taken: 54.326 seconds
```

Figure 60 : Remplissage de la table tf_ca (3)

Concernant le remplissage des tables, les figures montrent cette action suivant l'algorithme MapReduce.

Toutes les tables que nous venons de créer et remplir sont stockées dans le répertoire 'warehouse' de Hive. Nous illustrons par la figure suivante la table de fait tf_ca.

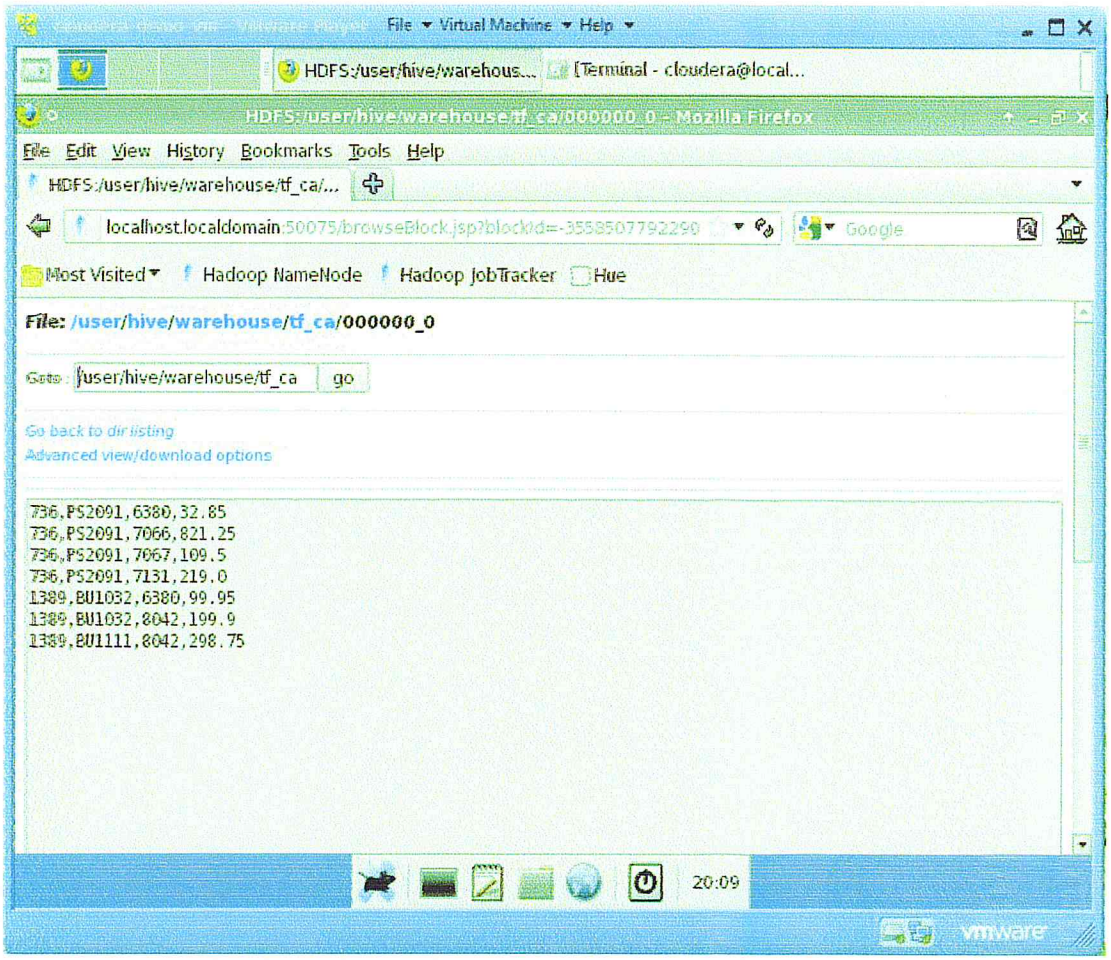

Figure 61 : La table tf_ca

3- Conclusion

De nos jours, la volumétrie des données ne cesse d'augmenter. Et vu la différence de qualité de ces données, le processus ETL est chargé de les trier, les transformer et les charger dans un entrepôt.

Dans ce chapitre, nous sommes sortis du relationnel en mettant en œuvre le processus ETL sous l'environnement Hadoop Hive qui dispose d'une implémentation complète du paradigme MapReduce afin d'accélérer les trois étapes de ETL.

Conclusion générale

Conclusion générale

L'expérience d'une entreprise est matérialisée par l'ensemble des données sur les flux et événements vécus par celle-ci. Cet ensemble est stocké sous forme de fichiers et bases de données. Le système décisionnel dispose d'un processus d'extraction, transformation et chargement (ETL) dont la fonction est la préparation de ces données à des fins d'analyse.

De nos jours, la volumétrie des données ne cesse de croitre rendant le travail du processus ETL complexe et lent en termes de temps.

Pour palier à ce problème, nous avons mis en œuvre un processus ETL sous l'environnement Hadoop Hive qui dispose d'une implémentation complète de MapReduce. Ce paradigme effectue des calculs parallèles et distribués des données potentiellement très volumineuses.

Dans ce mémoire, nous avons développé les chapitres suivants :

- Chapitre I : dans ce chapitre nous avons défini le système décisionnel et ses principales phases.
- Chapitre II : ce chapitre défini le processus ETL et ses trois étapes (extraction, transformation et chargement des données) ainsi que son importance et sa complexité. Vient ensuite le paradigme MapReduce qui est une solution pour accélérer le travail de ETL.
- Chapitre III : dans ce chapitre nous avons défini l'environnement Hadoop, ses sous projets MapReduse, Hive et HDFS.
- Chapitre IV : il englobe les trois premiers chapitres car il consiste en la mise en œuvre du processus ETL sous l'environnement Hadoop Hive. Pour cette mise en œuvre, nous avons créé une base de données sous MySQL, importé les tables de MySQL à HDFS en utilisant l'outil Sqoop qui permet des importations/exportations entre un SGBD tel que MySQL et Hadoop. Une fois les données dans HDFS, nous avons choisi un schéma en étoile et nous avons crée sa table de faits et ses dimensions avec les requêtes HiveQL qui s'exécutent en MapReduce.

Ce projet nous a permis de bien comprendre les systèmes décisionnels, l'importance du processus ETL et l'environnement Hadoop Hive qui intègre le MapReduce.

Durant ce travail, nous avons acquis de nouvelles connaissances informatiques grâce à l'utilisation de récents logiciels et nouveaux paradigmes conçus pour l'exploitation des données à forte intensité.

Nous souhaitons que les entreprises Algériennes adoptent l'environnement Hadoop pour un meilleur stockage et analyse de leur données. Cet environnement est largement utilisé aux Etas Unis et récemment en France.

Pour des travaux futures, nous proposons l'utilisation d'un grand nombre de sources de données (téraoctets voire pétaoctets) avec l'intégration des autres sous projets de Hadoop.

Références bibliographiques et webographiques

Références

[Akoka,Comyn-Wattiau, 2004] J. Akoka, I. Comyn-Wattiau, Systèmes d'information décisionnels, (2001).

[Chaudhuri, 1997] S. Chaudhuri, Data warehouse and OLAP for decision support, Microsoft research, Redmond: SIGMOD Record, USA(1997).

[Gruau,2004] C. Gruau, SQL Server 2000, Analysis Services et DTS, (2004).

[Imhoff,Galemmo,Geiger, 2003] C. Imhoff, N. Galemmo, J. Geiger, Mastering Data Warehouse Desing: Relational and Dimensional Techniques, Wiley Publishing (2003).

[Inmon, 2002] W. H. Inmon, Building the Data Warehouse Third Edition, Wiley Publishing (2002).

[Inmon, 2005] W. H. Inmon, Building the Data Warehouse Fourth Edition, Wiley Publishing (2005).

[Jaffre,Rauzy, 2010] M. Jaffre, P. Rauzy, MapReduce, (2010).

[Kimball, 2001] R. Kimball, Entrepôt des données, Guide Pratique de Construction Data Warehouse, WILEY (2001).

[Kimball, 2002] R. Kimball, M. Ross, ENTREPOT DE DONNEES Guide pratique de modélisation dimensionnelle, 2ème édition, WILEY (2002).

[kimball, 2004] R. Kimball, J.caserta, The Data Warehouse ETL Toolkit, Wiley Publishing (2004).

[Liu,Thomson,Pederson, 2011] X. Liu, C. Thomson, T. Pederson, ETLMR: A Highly Scalable Dimensional ETL Framework based on MapReduce, (2011).

[Noll, 2007] M. G. Noll, Running Hadoop On Ubuntu Linux(SingleNode Cluster), (2007).

[Ponniah, 2001] P. Ponniah, Data Warehousing Fundamentals: A Comprehensive Guide For IT Professionals, Wiley Publishing (2001).

[Reinschmidt,Francoise,2000] J. Reinschmidt, A. Francoise, Business Intelligence Certification Guide, IBM (2000).

[Reix, 2004] R. Reix, Systèmes d'information et management des organisations, (2004).

[Shanon,1975] R. E. Shanon, Systems Simulation the arts and science, Prentice Hall (1975).

[Sanjay,2008] R. Sanjay, HDFS Under The Hood, (2008).

[TANNIR, 2011] K. Tannir, MapReduce Algorithme de parallélisassions des traitements, (2011).

[White, 2009] T. White, Hadoop The Definitive Guide, O'REILLY, (2009).

Références bibliographiques et webographiques

[Web 1]
Introduction à l'informatique décisionnelle
http://www710.univ-lyon1.fr/~elghazel/Cours/SIAD/preambule.pdf

[Web 2]
http://www.opensides.fr/2011/03/10/hadoop-en-moins-de-5-minutes/

[Web 3]
http://hadoop.apache.org/

[Web 4]
http://hadoop.apache.org/common/docs/r0.20.2/hdfs_design.html

[Web 5]
https://cwiki.apache.org/confluence/display/Hive/Home

[Web 6]
http://sqoop.apache.org/

Remarque :
Dernière consultation des liens : Septembre 2012.